Hierarchical Visual Perception without Calibration

François Gaspard, Thierry Viéville

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Hierarchical visual perception without calibration

François Gaspard and Thierry Viéville

N° 3002

Octobre 1996

THÈME 3

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

Hierarchical visual perception without calibration

François Gaspard and Thierry Viéville

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet RobotVis

Rapport de recherche n ° 3002 — Octobre 1996 — 67 pages

Abstract: We have analyzed the equations and the formalism which allow to achieve dynamic visual perception of geometric and kinematic 3D information, for a monocular visual system without calibration.

Considering the emergence of active visual systems for which we **can not** consider that the calibration parameters are either known or fixed, we develop an alternative strategy based on the two complementary facts that:

(i) several perceptual tasks can be performed without knowing the calibration parameters,

while, for other perceptual tasks:

(ii) certain class of special displacements induce enough equations to evaluate the calibration parameters,

so that we can recover the Euclidean structure of the scene when needed.

A synthesis of what can be recovered in terms of scene geometry and kinematics is proposed. We give, for the different levels of calibration, an exhaustive list of the geometric and kinematic information which can be recovered.

Following a strategy based on special kind of displacements, such as fixed axis rotations or pure translations for instance, we also describe how to control the robotic system in order to generate these particular classes of displacement.

The implementation of these equations is analyzed here, and some experimental results are reported.

Key-words: Structure and Motion, Singular Displacements, Reconstruction

(Résumé : tsvp)

Perception visuelle hiérarchique sans calibration.

Résumé :

Considérant l'émergence de systèmes visuels actifs monoculaires, nous nous intéressons au formalisme et aux équations qui permettent d'obtenir une perception visuelle de la géométrie et de la cinématique d'une scène 3D, à partir d'un système visuel monoculaire non calibré.

Cependant l'utlisation de tels systèmes, ne nous permet plus de considérer que les paramètres de calibration sont connus ni même constants, on développe ici une stratégie alternative basée sur deux faits complémentaires:

(i) plusieurs tâches de perception peuvent être effectuées sans connaître les paramètres de calibration,

tandis que, dans le cas contraire:

(ii) des classes particulières de mouvement permettent de générer assez d'équations pour évaluer les paramètres de calibration,

permettant ainsi de récupérer -si besoin- la structure euclidienne de la scène observée.

Une synthèse de ce qui peut être estimé en terme d'attributs géométriques et cinématiques de la scène est proposée.

En particulier, nous décrirons les différents niveaux de calibration et donnerons une liste exhaustive des différentes informations obtenues à chaque niveau.

En suivant une stratégie basée sur certains types de déplacements, tels que -par exemple- des rotations autour d'axe fixe ou des translations pures, on décrit comment contrôler le système robotique de façon à générer de tels mouvements.

L'implémentation de ces équations est analysée ici et des résultats expérimentaux proposés.

Mots-clé : Structure et Mouvement, Mouvements Singuliers, Reconstruction

Contents

1 Introduction

The analysis of motion in the case of an un-calibrated monocular image sequence has already been developed by several authors [40, 17, 41, 38, 29, 46, 15] considering point and/or line correspondences or correspondences between planar patches and using either a discrete or a continuous representation of the rigid displacement between two or more frames. Considering these pieces of theory, and acting in the scope of 3D-active vision [8, 35], we would like to synthesize and integrate different mechanisms of active visual perception of object motion, scene structure and calibration in a comprehensive framework of sensing strategy, as realized in other fields of computer vision (see [31] for a review).

Contrary to historical studies in active vision [1, 3, 2] or more recent studies in the or more recent studies in the field [4, 5, 21, 20, 23, 8] we must not assume that the calibration parameters are known, while they may vary from one frame to another [7, 42, 16, 43, 8] so that calibration is now a part of the problem.

Knowing the calibration parameters of an active visual system is a hard job as soon as zoom, focus and vergence is used [8, 22, 16], and *we must not consider an active visual system is calibrated.*

Considering this fact, the key ideas of the present study are twofold:

- **Several perceptual tasks can be performed without knowing the calibration parameters.**

  For instance, the relative location of a point with respect to a plane [25], which applications are obstacle avoidance [45] or the computation of the convex envelop of an object [25], can be computed without recovering the Euclidean or affine calibration parameters.

- **Otherwise, singular displacements induce enough equations to evaluate the calibration parameters.**

  For instance, fixed axis rotations of known angles [34] or pure rotations [41] allow to estimate the calibration parameters, their uncertainty and, for a given kind of displacement, which parameters are optimally estimated [34], so that active visual strategies can be developed. On the other hand, pure translations do not allow to calibrate the Euclidean geometry of the scene [40], but its affine geometry [45].

Collecting all this information and considering a suitable statistical framework as in [10], it is then possible to infer which kind of displacement will increase at most the information (usually represented by the inverse of a covariance matrix) on the scene geometry, object kinematics and calibration parameters [6, 19].

This is the goal of our work.

In order to attain this objective, we are first going to review the theory of motion when no calibration: equations, parameterization of motion, etc...

We then are going to propose a synthesis of what can be recovered in terms of scene geometry and kinematics when calibration is not given as an input: describe the different forms of calibration, the different levels of calibration and give an exhaustive list of the different geometric and kinematic information to be recovered, depending the chosen geometry.

2 Revisiting the theory of motion when no calibration.

In this chapter we propose a short synthesis of the algebraic framework actually available for our purpose. Some of the equations have been given in a simpler form.

Notations.

We write vectors and matrices using bold letters, matrices being written with capital letters. The duals of vectors are represented as the transpose of a vector and scalars in italic. The notation $\mathbf{x} \wedge \mathbf{y} = \tilde{\mathbf{x}}\mathbf{y}$ corresponds to the cross-product, the dot-product being written as $\mathbf{x}^T\mathbf{y}$. $\tilde{\mathbf{x}}$ is a 3×3 skew-symmetric matrix[1] . The identity matrix is written $\mathbf{I}$. Geometric objects such as points, lines, planes are written with capital letters in 3D, and small letters in 2D. We represent the components of a matrix or a vector using superscripts from 0 to 2, e.g.: $\mathbf{x} = (x^0, x^1, x^2)^T$.
We write $\mathbf{a} \equiv \mathbf{b}$ if $\mathbf{a}$ is equal to $\mathbf{b}$ up to a scale factor, i.e. $\exists k, \mathbf{a} = k\,\mathbf{b}$.

2.1 Setting the equations

2.1.1 Camera model and frame of reference.

We use *the standard pinhole model* for a camera, assuming the camera performs a perfect perspective transform with center C (the camera optic center) at a distance f (the focal length) of the retinal plane. The pinhole model can still be used for a zoom lens if the object-to-image distance is not considered as fixed.

All coordinates are related to an affine frame of reference $\mathcal{R} = (C, \mathbf{x}, \mathbf{y}, \mathbf{z})$ *attached to the retina*, $\mathbf{z}$ being aligned with the optical axis, $\mathbf{x}$ and $\mathbf{y}$ being aligned with the horizontal and vertical axe in the image. The retinal plane is thus perpendicular to the optical axis Cz,as shown in figure 1.

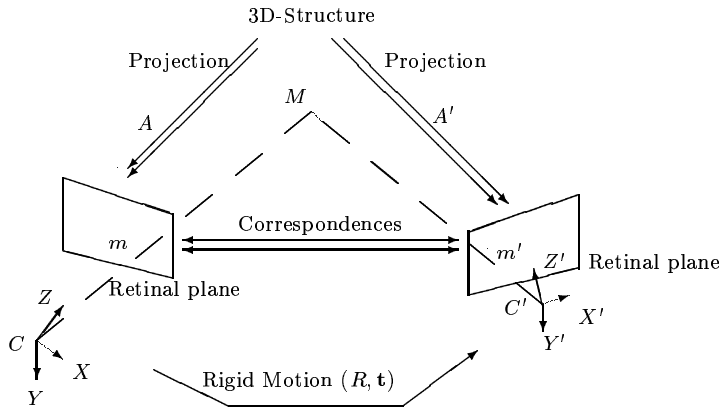

Figure 1: Elements used in the definition of the problem

[1]Remember that a 3×3 skew-symmetric matrix has 3 parameters and can always be represented by the crossproduct of a vector, i.e. is of the form $\tilde{\mathbf{x}}$ for some $\mathbf{x}$.

2.1.2 Using set of points as primitives.

We represent a 3D-point M by the vector $\mathbf{M} = C\vec{M} = (X, Y, Z)^T$ using Euclidean coordinates. Points in the retina, with pixel coordinates coordinates (u, v) will be represented as homogeneous 3-D vectors: $\lambda\ \mathbf{m} = \lambda\ C\vec{m} = \lambda\ (u, v, 1)^T$, corresponding to lines of a given direction passing through the optical center (2-D projective space).

Feature points corresponding to high curvature points are extracted from each image. In our application, we use the "Harris" corner detector [14], and as reported in [46], we perform a correlation operation and select those locations for which the correlation score is high.

When considering an image sequence, each point will be tracked from the first to the last view and then the trajectory will be interpolated using a polynomial model as proposed and implemented in [34] in order to obtain a sub-pixel mechanism of localization of the points.

We also represent geometric objects such as lines, planes or even conics (not implemented here) using set of points. For instance, line segments will be represented as a pair of points or a set of collinear points, planar structures a triplet of points or a set of coplanar points, etc...

Examples are shown in figure 2. For instance, if line segments are detected by the early-vision module they can be integrated in our system, using the endpoints.

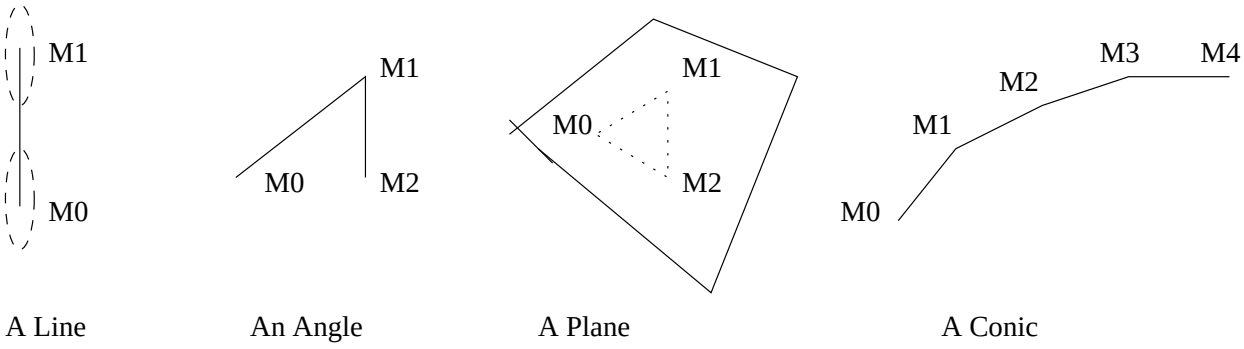

Figure 2: Defining geometric elements from set of points

Most of these points are "sliding" points in the sense that they can be replaced by similar points on the same object, but such an element can be integrated to represent lines or planes as discussed in [40, 41]. This notion of sliding points is well formalized by attaching a covariance matrix to each point, with a corresponding ellipsoid of uncertainty. Considering the line segment of figure 2 for instance, we see that $M0$ and $M1$ have an ellipsoid of uncertainty higher in the direction of the line segment, to have a model occlusions, early-vision errors, etc...

2.1.3 A suitable model of the intrinsic parameters of the camera.

In this study, *we do not assume the system is calibrated.* However, we are in a specific situation because we have chosen a "canonical" frame attached to the retina. Therefore, we consider only the matrix of the intrinsic parameters (called A-matrix) in the projection and write:

$$Z\,\mathbf{m} = \mathbf{A}\,\mathbf{M} \quad , \quad \mathbf{A} = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad with \quad \mathbf{a} = (u_0, v_0, f) \tag{1}$$

The vector $\mathbf{a}$ defines the intrinsic calibration parameters. A complete review can be found in [10].

In the present model, (u_0, v_0) is the principal point, and f the focal length; following [36], we assume that we know the ratio between the horizontal and vertical focal length and that we assume that the two retinal coordinates are orthogonal. It has been shown experimentally that these assumptions are valid for standard cameras [36] and also for high-level visual sensors [42]. Using this simple model will allow us to improve the obtained results.

We also assume that the intrinsic parameters are different for each camera position, as during a zoom. In the consecutive frame $\mathcal{R}' = (C', \mathbf{x}', \mathbf{y}', \mathbf{z}')$ we write:

$$Z'\,\mathbf{m}' = \mathbf{A}'\mathbf{M}' \quad with \quad \mathbf{a}' = (u'_0, v'_0, f') \tag{2}$$

2.1.4 Representation of rigid displacements.

We consider motion of rigid objects and the ego-motion of the camera, *in the discrete case.* We thus represent motion through rigid displacements.

It means that the tokens in the scene are undergoing a rigid displacement parameterized by a rotation matrix R and a translation vector $\mathbf{t}$:

$$\mathbf{M}' = \mathbf{R}\,\mathbf{M} + \mathbf{t} \tag{3}$$

2.2 Parameterization of motion when no calibration.

The goal of the parameterization of motion is the following: given a set of points in correspondence between two views, i.e. a set of matches $\{m.m'\}$ we want to analyze all constraints which relate the two points i.e find the equations of the form $\forall\{m.m'\}, f(m, m') = 0$. In particular, we would like to predict the location of a point given its correspondent, i.e. a relation of the form $\forall\{m.m'\}, m' = g(m)$. Having such parameterization allows to exact all information available from the retinal displacements, which is measured through the set of matches.

2.2.1 The Qs-representation and the F-matrix.

Considering the 2D correspondences between two points m and m' in two different frames, we obtain, combining equation (1),(2) and (3):

$$Z'\,\mathbf{m}' = Z\,\underbrace{\mathbf{A}'\mathbf{R}\mathbf{A}^{-1}}_{\mathbf{H}_\infty}\mathbf{m} + \underbrace{\mathbf{A}'\mathbf{t}}_{\mathbf{s}} \tag{4}$$

where the Q-matrix $\mathbf{H}_\infty$ corresponds to the "un-calibrated rotational component of the rigid displacement", or more geometrically *the collineation of the plane at infinity*, while the s-vector corresponds to the "un-calibrated translational component of the rigid displacement", also called "focus of expansion" by some authors, and more geometrically *the epipole*. These notations have been introduced in [40, 17] to analyze the motion of points and lines in the general case.

If we eliminate Z and Z' in equation (4) (by taking the cross-product with $\mathbf{s}$ and multiplying by $\mathbf{m}'^T$ we obtain:

$$\mathbf{m}'^T\,\underbrace{[\tilde{\mathbf{s}}\,\mathbf{H}_\infty]}_{\mathbf{F}}\,\mathbf{m} = 0 \tag{5}$$

The matrix $\mathbf{F} = \tilde{\mathbf{s}}\ \mathbf{H}_\infty = \mathbf{A}'^{-1T}\,\tilde{\mathbf{t}}\,\mathbf{R}\,\mathbf{A}^{-1}$ is the *Fundamental matrix* and is also called the "essential matrix in the un-calibrated case". If we consider that the only information available is related to the retinal correspondences between points, without any knowledge about the depths Z, equation (5) is the only equation that can be derived [40].

Considering a set of matches related by equation (3) the equation (5) is well defined if and only if (i) $\mathbf{s} \neq 0$ and (ii) there is no linear relations between all m' and m. The degenerated cases occur only if the translation is zero, or if all points belong to the same plane [40][2]. This particular case will be analyzed in detail. Reciprocally, as soon as the points belongs to at least two planes, we can defined a F-matrix [18].

A camera for which F has been computed is called *a weak calibrated camera.*

Let us make an important remark here. Without any knowledge of the calibration parameters, we can detect a singular situation: *the case where the translation* $\mathbf{t}$ *is parallel to the retinal plane*, i.e. $t^2 = 0$ since we have $t^2 = 0 \equiv s^2 = 0$. This singular situation corresponds to the geometrical fact that the epipole are at infinity and is of practical interest.

2.2.2 Depth from motion equations.

Two other equations derived from equation (4) allow to reconstruct the 3-D-depths Z and Z'.

[2]From algebraic point of view, equation (5) has singular solutions if and only if there exist a linear relation between $\mathbf{m}$ and $\mathbf{m}'$, i.e. a relation of the form $\mathbf{m}' = \mathbf{H}\ \mathbf{m}$. This situation corresponds to the case were the points are related by a collineation, i.e. correspond to a planar structure as reviewed in the sequel.

1. The *depth from motion equation* (obtained taking the cross-product with $\mathbf{m}'$ and multiplying by $\mathbf{s}^T$) in the un-calibrated case:

$$\frac{||\mathbf{s}||}{Z} + (\mathbf{r}^T\mathbf{m}) = \frac{||\mathbf{F}\mathbf{m}||}{||\mathbf{s}||}\underbrace{\left[\frac{(\mathbf{s}^T\mathbf{m}')}{||\mathbf{s}\wedge\mathbf{m}'||}\right]}_{cot(\widehat{\mathbf{s}\mathbf{m}'})} = \pi_{\mathbf{m}} \tag{6}$$

   where $\mathbf{r} = \mathbf{H}_\infty^T \frac{\mathbf{s}}{||\mathbf{s}||}$ is an unknown quantity related to the affine calibration of the scene [40] and the inverse of the depth or proximity for an image point is directly related to the F-matrix up to a linear function of the retinal location. Moreover, $||\mathbf{s}||$ corresponds to an unknown scale factor, as always present in monocular systems.

   Note that $\pi_{\mathbf{m}}$, called the *projective retinal proximity* in the sequel, is given in pixel, as visible in equation (15) for instance.

   Furthermore, the quantity:

$$\frac{||\mathbf{s}||}{Z} = \underbrace{\pi_{\mathbf{m}} - (\mathbf{r}^T\mathbf{m})}_{\delta_{m,\mathbf{H}_\infty}} \geq 0 \tag{7}$$

   called the *affine retinal proximity*, also in pixel allows to decide :

   - Whether points are at the "horizon", i.e. at an infinite distance or a negligible proximity, if $\delta_{m,\mathbf{H}_\infty} \simeq 0$. As discussed in [37] for the calibrated case and [41] for the un-calibrated case, these points only have a rotational disparity.
   - Whether a points is behind another (relative depth) since, $m_1 \quad behind \quad m_2 \Leftrightarrow \delta_{m_1,\mathbf{H}_\infty} < \delta_{m_2,\mathbf{H}_\infty}$. Note that relative depth is an affine attribute and is undefined for projective geometry.

2. The *depth evolution equation* (obtained taking the cross-product with $\mathbf{s}$ and the norm of the result) relates the depth from one frame to another:

$$\frac{\frac{1}{Z}}{\frac{1}{Z'}} = \frac{||\mathbf{F}\,\mathbf{m}||}{||\mathbf{s}\wedge\mathbf{m}'||} = \nu_{\mathbf{m}} \tag{8}$$

In fact, it can be easily shown that equations (5), (6) and (8) are equivalent to equation (4), since $\mathbf{F}\mathbf{m} \equiv \mathbf{s}\wedge\mathbf{m}'$, while these two vectors are in the same direction.

2.2.3 The case of a pure rotation, and the planar case.

As pointed out previously, in the case of a pure rotation or if the set of points belongs to a unique planar structure, we cannot estimate the F-matrix because

all points in one view are related to points in the other view by a relation of the form:

$$\mathbf{m}' \equiv \mathbf{H}\,\mathbf{m} \tag{9}$$

This corresponds to two equations for each match.

Let us consider the plane normal is $\mathbf{n}$, with $||\mathbf{n}|| = 1$ and $d > 0$ its distance to the origin. They define the equation of the plane $\mathcal{P}$, i.e. $M \in \mathcal{P} \Leftrightarrow \mathbf{n}^T\, M = d$ and we can write:

$$\mathbf{H} = \mathbf{A}' \left[\mathbf{R} + \mathbf{t}\frac{\mathbf{n}^T}{d}\right] \mathbf{A}^{-1} = \mathbf{H}_\infty + \mathbf{s}\,\nu^T \quad with \quad \nu = \mathbf{A}^{-1\,T}\frac{\mathbf{n}^T}{d} \tag{10}$$

Geometrically the case of a pure rotation is strictly similar to the case of an unique plane, the plane being the plane at infinity, i.e. $\mathbf{H} = \mathbf{H}_\infty$, with $1/d = 0$.

If we assume that we have been able to estimate a F-matrix and now want to analyze a planar structure which rigid motion is compatible with $\mathbf{F}$ [41], we have the following 5 dimensional[3] constraint :

$$\mathbf{F}\,\mathbf{H}^T + \mathbf{H}\,\mathbf{F}^T = 0 \Leftrightarrow \tilde{\mathbf{s}}\,\mathbf{H} = \mathbf{F} \tag{11}$$

More precisely, such a collineation can be represented by a vector $\mathbf{h} = \mathbf{H}^T \frac{\mathbf{s}}{||\mathbf{s}||}$, i.e. 3 parameters, with:

$$\mathbf{H} = -\frac{\tilde{\mathbf{s}}}{||\mathbf{s}||^2}\mathbf{F} + \frac{\mathbf{s}}{||\mathbf{s}||}\,\mathbf{h}^T \tag{12}$$

For a given point $\mathbf{m}$, we have, from (6) and (12):

$$\nu_{\mathbf{m}}\,\mathbf{m}' = \mathbf{H}\,\mathbf{m} + \underbrace{\left[\pi_{\mathbf{m}} - \left(\mathbf{m}^T\,\left(\mathbf{H}^T\,\frac{\mathbf{s}}{||\mathbf{s}||}\right)\right)\right]}_{\delta_{\mathbf{m},\mathbf{H}}}\,\frac{\mathbf{s}}{||\mathbf{s}||} \tag{13}$$

so that the displacement of a given point corresponds to the collineation if and only if $\delta_{\mathbf{m},\mathbf{H}} = 0$.

This last equation has three consequences:

- Given three non collinear points and a F-matrix, it is always possible to compute the collineation of the plane defined by these three points, considering that $\delta_{\mathbf{m},\mathbf{H}} = 0$ for these three points.
- It is possible to determine if 4 points or more are collinear, by verifying if the set of equations obtained using (13) for each point is of rank 3, i.e. if a suitable set of determinants vanish.

[3] A symmetric 3×3 matrix is equal to zero if and only of its 6 components vanish, but they are here defined up to a scale factor.

- Choosing a orientation of the 3D-space, a point is "behind" the plane if and only if $\delta_{\mathbf{m},\mathbf{H}} < 0$ and "in front of the plane" if and only if $\delta_{\mathbf{m},\mathbf{H}} > 0$ [41, 25].

  As a consequence we can infer the relative position of a point with respect to a plane using this equation.

On the reverse let us consider that we have estimated a planar displacement defined by a matrix $\mathbf{H}$. Given two other points not in the plane we can estimate the vector $\mathbf{s}$ since from equation (13) we have $\mathbf{s} \perp (\mathbf{m}' \wedge (\mathbf{H}\ \mathbf{m}))$, while we can estimate the matrix $\mathbf{F}$ from equation (11). It is thus very easy to estimate a rigid displacement given a planar patch and two additional points.

2.2.4 Reduction of the motion equation

If we consider *any collineation* $\mathbf{H}$ *compatible with a F-matrix* $\mathbf{F}$, combining equations (5) and (11) we obtain the *reduced motion equation*:

$$\mathbf{m}'^{T}(\mathbf{s} \wedge \mathbf{m}^*) = 0 \tag{14}$$

with $\mathbf{m}^* \equiv \mathbf{H}\ \mathbf{m}$, since we have: $\mathbf{s} \wedge \mathbf{m}^* \equiv \tilde{\mathbf{s}}\ \mathbf{H}\ \mathbf{m} \equiv \mathbf{F}\mathbf{m}$.

In other words, considering $\mathbf{m}^*$ instead of $\mathbf{m}$, it is as if we were in the case of pure translation, with $\mathbf{H}_\infty = \mathbf{I}$, i.e. neither rotational motion, nor variation of the calibration parameters. This simplification is possible, because these two last effects only correspond to a transformation of the image plane and are not related to the observed scene [10, 41]. This mechanism is equivalent to the image rectification in the stereo paradigm [10].

Among all possible collineations $\mathbf{H}$, we can choose the one which minimizes the image deformation, i.e. having a set of matches which minimizes the average retinal disparity, as defined in the sequel.

2.2.5 Predicting the pixel location in the next view.

Given two views, we can estimate the projective depth from equation (6) and propagate this estimation in the next view using equation (8). From this last information we can predict the new retinal location. More precisely, from equations (6), (3) and (12) we obtain:

$$\nu_{\mathbf{m}}\, m' = \frac{(\mathbf{F}\ \mathbf{m}) \wedge \mathbf{s}}{||\mathbf{s}||^2} + \pi_{\mathbf{m}}\, \frac{\mathbf{s}}{||\mathbf{s}||} \tag{15}$$

This can be used in tracking processes or -as discussed now- in order to fuse information along an image sequence.

2.2.6 Dealing with several displacements

Let us now assume that several rigid objects with different displacements are present in the scene.

The problem arises here that several rigid displacements may correspond to the same matrix:

$$\mathbf{F} \equiv \mathbf{A}^{'-1\,T} \underbrace{\left[\tilde{\mathbf{t}}\,\mathbf{R}\right]}_{\mathbf{E}} \mathbf{A}^{-1} \tag{16}$$

First of all, if two rigid displacements differ only from the magnitude of their translation $||\mathbf{t}||$ they yield the same F-matrix. It is known that, if $\mathbf{A} = \mathbf{A}' = \mathbf{I}$ this is the only ambiguity [10], i.e. that there is a unique decomposition of $\mathbf{E}$ if this matrix verifies the Huang-Faugeras conditions. If the A-matrices are not equal to the identity it has been demonstrated [18] that the two constraints on the calibration parameters, the Kruppa equations, are equivalent to the two Huang-Faugeras conditions, so that as soon as these equations are verified we obtain a matrix of the form $\mathbf{E}$ with one solution for the rigid displacement (up to an indetermination of the magnitude of $\mathbf{t}$).

This thus really allows to distinguish two rigid displacements up to an indetermination of the magnitude of $\mathbf{t}$. This method has been developed and experimented in [33],using a particular form for the F-matrix.

However, if we consider the literature on the subject [30, 27, 33, 32, 12, 20, 44, 13] all authors implicitly or explicitly segment the image considering that the scene is a set of planar patches.

This idea simply means that we do not consider the relief of the object we want to detect, whereas we simplify the model and consider it as being flat. This seems to be enough to detect whether it is in motion or not, while computing its relief is another problem.

Therefore, similarly as before, the problem arises that several rigid planar displacements may correspond to the same matrix:

$$\mathbf{H} \equiv \mathbf{A}^{'} \underbrace{\left[\mathbf{R} + \mathbf{t}\,\frac{\mathbf{n}^T}{d}\right]}_{\mathbf{G}} \mathbf{A}^{-1} \tag{17}$$

First of all, if two rigid displacements differ only from the magnitude or sign of their translation, they yield the same H-matrix, considering two planes with the same orientation but different distances to the origin. It is also known that, if $\mathbf{A} = \mathbf{A}' = \mathbf{I}$ this is the only ambiguity [10]. Now, in the un-calibrated case, if we consider two collineations $\mathbf{H}_1 \equiv \mathbf{A}^{'}\,\mathbf{H}_1^c\,\mathbf{A}^{-1}$ and $\mathbf{H}_2 \equiv \mathbf{A}^{'}\,\mathbf{H}_2^c\,\mathbf{A}^{-1}$ it is clear that $\mathbf{H}_1 \equiv \mathbf{H}_2 \Leftrightarrow \mathbf{G}_1 \equiv \mathbf{G}_2$, but $\mathbf{G}_1$ and $\mathbf{G}_2$ corresponds to the calibrated case, and are thus equivalent if and only if they correspond to the same rigid displacement up to the magnitude of the translation.

Finally, considering a rigid displacement represented by a F-matrix and a planar rigid displacement represented by a H-matrix, it has also been established

that these two parameterization corresponds to the same rigid displacement up to the magnitude of the translation, if and only if equation (11) is verified [41].

This shows that our equations allows to distinguish between two different rigid displacements in any case. Moreover, very often, the retinal displacement is taken as a retinal displacement [12, 33, 20, 32, 13] because this model of 2D-displacement is quite robust to detect, and is sufficient in most experimental situations. As for the relief of the object, none of these approaches attempt to explicitly recover the rigid motion or the 3D-structure of the detected moving objects.

2.3 Calibration and reconstruction.

Considering recent contributions on singular displacements analysis we now develop how such configurations allow to solve calibration and reconstruction problems. Specific contributions have been given in a second part.

2.3.1 Euclidean, projective and affine reconstruction.

We can reconstruct the Euclidean position $\mathbf{M}$ of a point $\mathbf{m}$, using equations (1) and (6) from:

$$\mathbf{M} = ||\mathbf{s}|| \frac{\mathbf{A^{-1}\,m}}{\pi_{\mathbf{m}} - (\mathbf{r}^T\mathbf{m})} \tag{18}$$

but, in our case, we do not know $\mathbf{A}$, $\mathbf{r}$ and $||\mathbf{s}||$, and the reconstruction is defined up to the 5 parameters of $\mathbf{A}$, the 3 parameters $\mathbf{r} = \mathbf{H}_\infty^{\mathbf{T}}\,\frac{\mathbf{s}}{\|\mathbf{s}\|}$ and a scale-factor $||\mathbf{s}||$.

If we define the particular reconstruction obtained for $\mathbf{A} = \mathbf{I}$ and $\mathbf{r} = 0$ and $||\mathbf{s}|| = 1$, i.e. $\mathbf{M_\bullet} = \frac{\mathbf{m}}{\pi_{\mathbf{m}}}$ the reconstructed point $\mathbf{M}$ is related to $\mathbf{M_\bullet}$ by the following relation, obtained from (18):

$$\mathbf{M_\bullet} = \frac{\mathbf{m}}{\pi_{\mathbf{m}}} \Rightarrow \mathbf{M} = ||\mathbf{s}|| \frac{\mathbf{A^{-1}M_\bullet}}{1 - (\mathbf{r}^T\mathbf{M_\bullet})} \tag{19}$$

which is a particular projective transform of the 3D-space, parameterized by $||\mathbf{s}||$, $\mathbf{A}$ and $\mathbf{r}$.

If we know the affine structure of the scene, i.e. $\mathbf{r}$, we can define similarly:

$$\mathbf{M}_\star = \frac{\mathbf{m}}{\pi_{\mathbf{m}} - (\mathbf{r}^T\mathbf{m})} \Rightarrow \mathbf{M} = ||\mathbf{s}||\,\mathbf{A^{-1}M}_\star \tag{20}$$

which is related to the Euclidean structure of the scene up to a particular affine transform of the 3D space, given by $\mathbf{A}$.

The global scale factor $||\mathbf{s}||$ cannot be recovered since we are using a monocular system.

2.3.2 Projective self-calibration

The projective calibration consists in computing the fundamental matrix $\mathbf{F}$. Following [39] we can either compute this matrix in the case of a general rigid displacement or either in the case of some particular displacements (see 2.3.7).

From the fundamental matrix, we are able to estimate the epipole $\mathbf{s}$ as the null vector of $\mathbf{F}$ as $(\tilde{\mathbf{s}}\,\mathbf{H}_\infty)\,\mathbf{s} = \mathbf{0}$. Thus having $\mathbf{F}$ and $\mathbf{s}$, we can compute (using equation (6)) the retinal proximity $\pi_{\mathbf{m}}$.

2.3.3 Affine self-calibration

In the general case we can't compute the collineation of the plane at infinity, i-e the affine calibration. Several authors use parallel lines to estimate $\mathbf{H}_\infty$ but it is not easy to know where are such parallel lines.

In our case where we want to estimate the structure of the scene, we can't assume that we know such cues. In the case of some particular displacements, we can simplify equation (3) and compute $\mathbf{H}_\infty$:

- In the case of a rotation, all the point in the scene are related, from one view to the other, by an unique collineation : $\mathbf{H}_\infty$.

  However we must note here that we can't estimate if the displacement is a rotation or if we are looking at a unique planar structure. We can suppose here to get this information from the vision system which can just specify if the displacement is a rotation or another one. Thus we can detect this particular displacement and compute this collineation.

- Another known and widely used case, is the case of a pure translation, with constant intrinsic parameters. As $\mathbf{A} = \mathbf{A}'$ and $\mathbf{R} = \mathbf{I}$, we have simply $\mathbf{H}_\infty = \mathbf{I}$. We can note here that translations often occurs but when we use active visual systems, we can't always assume we are in this case of a **pure** translation.

2.3.4 Euclidean self-calibration: general case

Let us write $\mathbf{K} = \mathbf{A}\mathbf{A}^T$ and $\mathbf{K}' = \mathbf{A}'\mathbf{A}'^T$. Each K-matrix is in one-to-one correspondence with the A-matrix[4] [40].

If we explicit the fact that R is an orthogonal matrix in the definition of equation (4), we obtain:

$$\mathbf{K}' \equiv \mathbf{H}_\infty\,\mathbf{K}\,\mathbf{H}_\infty^T \tag{22}$$

[4] In our case we have:

$$\left\{ \begin{array}{ccc} u_0 = K_{02} & v_0 = K_{12} & f^2 = K_{00} - K_{02}^2 = K_{11} - K_{12}^2 \\ with & K_{22} = 1 & K_{01} = K_{02}K_{12} \end{array} \right. \tag{21}$$

so that the six elements of the *symmetric* matrix $\mathbf{K}$ verifies 2 quadratic constraints, 1 linear constraint and allow to estimate the parameters of the corresponding A-matrix.

which leads to (multiplying left and right by $\tilde{\mathbf{s}}$):

$$\tilde{\mathbf{s}}\ \mathbf{K}'\ \tilde{\mathbf{s}} \equiv \mathbf{F}\ \mathbf{K}\ \mathbf{F}^T \tag{23}$$

which corresponds to *two independent* equations known as the Kruppa equations. These equations can be easily made explicit considering the singular-value decomposition of $\mathbf{F}$ given in equation (38). After some algebra, equation (22) reduces to:

$$\left[\begin{array}{l} \mathbf{u}_1^T\ \mathbf{K}'\mathbf{u}_1 \\ \mathbf{u}_1^T\ \mathbf{K}'\mathbf{u}_2 \\ \mathbf{u}_2^T\ \mathbf{K}'\mathbf{u}_2 \end{array}\right] \wedge \left[\begin{array}{ll} (\sigma_1)^2 & \mathbf{v}_1^T\ \mathbf{K}\mathbf{v}_1 \\ \sigma_1\ \sigma_2 & \mathbf{v}_1^T\ \mathbf{K}\mathbf{v}_2 \\ (\sigma_2)^2 & \mathbf{v}_2^T\ \mathbf{K}\mathbf{v}_2 \end{array}\right] = 0 \tag{24}$$

which, as a cross-product, is equivalent to 2 equations. They yield quartic equations in the intrinsic parameters not easily calculable [11] which are the only equations we can obtain for self-calibration in the general case.

These equations have an important negative consequence : *in the general case (non-constant calibration parameters and only projective data) it is not possible to self-calibrate a system*, even with our simple model of equation (1).

We must introduce another assumption, such as the fact that these parameters are constant [18] or discuss the calibration using another mechanisms, as developed now.

2.3.5 Euclidean self-calibration: using the affine calibration.

In the case where the calibration parameters are constant, and knowing $\mathbf{H}_\infty$ we can easily calculate the intrinsic calibration parameters since we have:

$$\mathbf{Q} = \frac{\mathbf{H}_\infty}{det(\mathbf{H}_\infty)^{\frac{1}{3}}} \quad and \quad \rho = \frac{\mathbf{A}\mathbf{u}}{det(\mathbf{A})} \quad \Rightarrow \mathbf{O} = \frac{\mathbf{Q} - \mathbf{Q}^{-1}}{2} = \sin(\theta)\ \mathbf{K}\ \tilde{\rho} \tag{25}$$

where $\mathbf{u}$ is the unary vector of the axis of the rotation and θ the angle of the rotation. In the general case we have $det(\mathbf{O}) = trace(\mathbf{O}) = 0$ and we can recover the intrinsic parameters only after two rotations not around the same axis [40].

We also recover ρ up to a scale factor, as the null vector of $\mathbf{O}$ since we have : $\mathbf{O}\,\rho = 0$.

With the model of equation (1) we obtain the following equations:

$$\left\{\begin{array}{rcl} O_{00} - u_0\,O_{20} & = & 0 \\ O_{11} - v_0\,O_{21} & = & 0 \\ (O_{01} + O_{10}) - u_0\,O_{21} - v_0\,O_{20} & = & 0 \\ O_{20}\,f^2 + O_{00}\,u_0 + O_{01}\,v_0 + O_{02} & = & 0 \\ O_{21}\,f^2 + O_{10}\,u_0 + O_{11}\,v_0 + O_{12} & = & 0 \end{array}\right. \tag{26}$$

which allow to recover (u_0, v_0, f) after one rotation except if:

$$O_{20} = -\frac{sin(\theta)}{f}u^1 = 0 \quad and \quad O_{21} = \frac{sin(\theta)}{f}u^0 = 0 \tag{27}$$

This singular configuration corresponds to a rotation around the optical axis only.

As for the projection of the translation $\mathbf{s}$ we can detect if the rotation axis is parallel to the retinal plane since $u^2 = 0 \Leftrightarrow \rho^2 = 0$.

2.3.6 Propagating the calibration.

Let us consider equations (46) and (22) again. These equations have two major consequences:

- As soon as the calibration parameters. i.e. $\mathbf{r}$ and $\mathbf{K}$ are known in one view, their estimation can be propagated in all the image sequence using equation (46) for $\mathbf{r}$ and then, knowing $\mathbf{r}$, using equation (22) for $\mathbf{K}$.

  Of course, the precision of this estimation will decrease with time, since errors will accumulate, but it allows to calibrate at a given instant and maintain this information for a certain period of time.

- The scale factor indetermination is unique for the whole sequence since we can estimate $||\mathbf{s}'||/||\mathbf{s}||$ between each consecutive displacements.

We must discuss one point here. In fact, knowing the F-matrices between each consecutive frames but also between a frame and the frame before the last frame we can directly estimate $\mathbf{r} - \mathbf{r}'$ and $||\mathbf{s}'||/||\mathbf{s}||$ [40]. However, in this case, we must not only consider pairs of views but also triplets of views, which is more heavy to manage. On the contrary, the new method introduced here, collect all information about the last before last frame in $\pi'_{\mathbf{m}'}$, and works directly on the data from linear equations. This should be a more efficient approach.

2.3.7 Using specific displacements.

As detailed in [39] there are several situations for which the F-matrix or the H-matrix have a particular interesting form. Considering a robotic system, it is very often that a displacement is not a general displacement but a constrained motion such as a pure translation, a fixed axis rotation of known angle, etc...

The related constraints are far from having negative properties. On the contrary, they induce additional equations which help solving the reconstruction or calibration problem. Furthermore, the estimation of the displacement are easier in these cases, because we have to evaluate less parameters.

Following [39], let us collect all these constraints.

Considering a rigid structure, the following class of displacements can be identified, N is the number of parameters:

Class of Displacement	Parameterization or constraint	Information Recovered	N
Pure rotation	$\mathbf{F}=0$	$\mathbf{H}_\infty, \mathbf{t}=0$	0
Z-axis pure translation	$\mathbf{F}=\begin{pmatrix}0&-1&0\\1&0&0\\0&0&0\end{pmatrix}$	$\mathbf{H}_\infty=\mathbf{R}=\mathbf{I}, \mathbf{t}\equiv\mathbf{z}$	0
Pure retinal translation	$\mathbf{F}=\begin{pmatrix}0&0&a\\0&0&b\\-a&-b&0\end{pmatrix}, \|\|\mathbf{F}\|\|=1$	$\mathbf{H}_\infty=\mathbf{R}=\mathbf{I}, \quad \mathbf{t}\equiv\begin{pmatrix}\cos(\theta)\\\sin(\theta)\\0\end{pmatrix}$ $\theta=atan(\frac{s1}{s0})$	1
Pure translation	$\mathbf{F}=\begin{pmatrix}0&c&a\\-c&0&b\\-a&-b&0\end{pmatrix}, \|\|\mathbf{F}\|\|=1$	$\mathbf{H}_\infty=\mathbf{R}=\mathbf{I}$	2
Retinal displacement	$\mathbf{F}=\begin{pmatrix}0&0&a\\0&0&b\\c&d&e\end{pmatrix}, \|\|\mathbf{F}\|\|=1$	$\mathbf{R}, \mathbf{t}/\|\|\mathbf{t}\|\|, eq(\mathbf{a})$	4
Zoom displacement	$\mathbf{F}=\begin{pmatrix}0&f&a\\-f&0&b\\c&d&e\end{pmatrix}, \quad \begin{matrix}c\,b-a\,d=0\\\|\|\mathbf{F}\|\|=1\end{matrix}$	$\mathbf{R}=\mathbf{I}, \mathbf{t}/\|\|\mathbf{t}\|\|, eq(\mathbf{a})$	4
Fixed axis rotation	$det(\mathbf{F}+\mathbf{F}^T)=0, \mathbf{Fs}\neq 0, det(\mathbf{F})=0, \|\|\mathbf{F}\|\|=1$	$eq(\mathbf{a})$	6
Rocket displacement	$\mathbf{Fs}=0, det(\mathbf{F})=0, \|\|\mathbf{F}\|\|=1$	$eq(\mathbf{a})$	6
Quarter turn rocket displacement	$\mathbf{F}=\mathbf{F}^T, det(\mathbf{F})=0, \|\|\mathbf{F}\|\|=1$	$eq(\mathbf{a})$	5
Retinal translation	$det(\mathbf{F})=0, s^2=0, \|\|\mathbf{F}\|\|=1$	$\mathbf{t}\equiv\begin{pmatrix}\cos(\theta)\\\sin(\theta)\\0\end{pmatrix}$ $\theta=atan(\frac{s1}{s0})$ $, eq(\mathbf{a})(Kruppa)$	6
General rigid displacement	$det(\mathbf{F})=0, \|\|\mathbf{F}\|\|=1$	$eq(\mathbf{a})(Kruppa)$	7

where $eq(\mathbf{a})$ means that we obtain equations about the intrinsic calibration parameters, these equations being either linear equations or the quartic Kruppa equations, as specified. In these cases, it is not possible to maintain an estimation of all calibration parameters.

In the planar case, we have:

Class of Displacement	Parameterization or constraint	Information Recovered	Number of Parameters
Stationary structure	$\mathbf{H}=\mathbf{I}$	$\mathbf{R}=\mathbf{I}, \mathbf{t}=0, \mathbf{a}=\mathbf{a}'$	0
Constant retinal displacement	$\mathbf{H}=\begin{pmatrix}0&0&a\\0&0&b\\0&0&1\end{pmatrix}$	$\mathbf{R}=I, \mathbf{t}\equiv(a,b,0), \mathbf{a}=\mathbf{a}', \mathbf{n}\equiv\mathbf{z}$	2
Retinal planar zoom	$\mathbf{H}=\begin{pmatrix}c&0&a\\0&c&b\\0&0&1\end{pmatrix}$	eq($\mathbf{a}$)	4
Retinal planar rotation	$\mathbf{H}=\begin{pmatrix}c&d&a\\-d&c&b\\0&0&1\end{pmatrix}$	$\mathbf{R}$, eq($\mathbf{a}$)	4
Pure planar retinal translation	$\mathbf{H}=\mathbf{I}+\mathbf{s}\nu^T, s^2=1$	$\mathbf{R}=\mathbf{I}, \mathbf{s}/\|\|\mathbf{s}\|\|, \nu$	5
Pure planar translation	$\mathbf{H}=\mathbf{I}+\mathbf{s}\nu^T, s^2=0$	$\mathbf{R}=\mathbf{I}, \mathbf{s}/\|\|\mathbf{s}\|\|, \nu$	5
Retinal planar displacement	$\mathbf{H}=\begin{pmatrix}a&b&c\\d&e&f\\0&0&1\end{pmatrix}$	$\mathbf{R}$, $\mathbf{t}/\|\|\mathbf{t}\|\|$, $\mathbf{n}$, eq($\mathbf{a}$)	6
General planar displacement	$\mathbf{H}=\begin{pmatrix}a&b&c\\d&e&f\\g&h&1\end{pmatrix}$		8

In fact some other variants have also been introduced in order to have alternative models with very few parameters. For instance a model with zero parameters, corresponding to a collineation equal to the identity, i.e. a stationary structure is introduced. This allows to have a simple model assuming that points are not moving.

2.3.8 Conclusion: collecting all calibration parameters.

Considering equations (5), (12) and (11) we have the following direct computations:

- We obtain directly $\mathbf{s}$ from $\mathbf{F}$, for any rigid displacement.
- We obtain directly $\mathbf{h}$ from $\mathbf{H}$ and $\mathbf{F}$, if a planar displacement is compatible with a rigid displacement.
- We obtain directly $\mathbf{F}$ from $\mathbf{H}$ and $\mathbf{s}$ of a rigid displacement is obtained from two rigid displacements.

so that we can complete the estimation of the displacement parameters as soon as we obtain some estimates of them.

Furthermore, with the previous equations, the following direct computations are available, from equations (25), (26) and (4):

- We obtain directly $\rho, \mathbf{r}, \mathbf{a}, \mathbf{R}, \mathbf{t}$ from $\mathbf{H}_\infty$ and $\mathbf{F}$.
- We obtain directly $\rho, \mathbf{a}, \mathbf{R}$ from $\mathbf{H}_\infty$.
- We obtain directly $\mathbf{t}$ from $\mathbf{a}$ and $\mathbf{F}$.
- We obtain directly $\mathbf{H}_\infty$ from $\mathbf{r}$ and $\mathbf{F}$.

Moreover, considering a previous estimation of the calibration parameters, we have the following feed-forward estimation:

- Using equation (46) and an estimation of $\mathbf{F}$ we obtain $\mathbf{r}$ and $\frac{||\mathbf{s}||}{||\mathbf{s}'||}$ from the previous estimation of $\mathbf{r}'$.
- Using equation (22) and an estimation of $\mathbf{F}$ and $\mathbf{r}$ thus $\mathbf{H}_\infty$ we obtain $\mathbf{a}$ from a previous estimation of $\mathbf{a}'$.

3 A few improvements of this state of the art

3.1 Reconstruction using the depth from motion equations

As described previously, to reconstruct depth, we use the depth from motion equations which allow to estimate the depth from the correspondences. Therefore, the equation used to compute depth must be as stable as possible. Let us now analyze this point.

If we take the cross product with $\mathbf{s}$ in equation (2), we obtain:

$$Z' (\mathbf{s} \wedge \mathbf{m}') = Z (\mathbf{s} \wedge \mathbf{H}_\infty \mathbf{m})$$

Using equation (5), we have:

$$(\mathbf{s} \wedge \mathbf{m}') = \frac{Z'}{Z}\mathbf{F}\ \mathbf{m} \tag{28}$$

We remark here that $\frac{Z'}{Z}$ is positive and the two vectors $(\mathbf{s} \wedge \mathbf{m}')$ and $\mathbf{F}\ \mathbf{m}$ are collinear and have the same direction.

Taking the cross-product, in equation (2) with $\mathbf{m}'$ and multiplying by $\mathbf{s}^T$, we obtain:

$$(\mathbf{m}' \wedge \mathbf{H}_\infty\ \mathbf{m}) + \frac{(\mathbf{m}' \wedge \mathbf{s})}{Z} = 0 \tag{29}$$

Since $\tilde{\mathbf{s}}\ \mathbf{H}_\infty = \mathbf{F}$, we have:

$$\mathbf{H}_\infty = -\frac{\tilde{\mathbf{s}}\ \mathbf{F}}{\|\ \mathbf{s}\ \|^2} + \frac{\mathbf{s}\ \mathbf{r}^T}{\|\ \mathbf{s}\ \|}$$

where $\mathbf{r} = {\mathbf{H}_\infty}^T \frac{\mathbf{s}}{\|\mathbf{s}\|}$.

By replacing $\mathbf{H}_\infty$ in equation (29), we obtain:

$$-\frac{\mathbf{m}' \wedge \mathbf{s} \wedge \mathbf{F}\ \mathbf{m}}{\|\ \mathbf{s}\ \|^2} + \frac{\mathbf{m}' \wedge \left(\mathbf{s}\ \mathbf{r}^T\right) \mathbf{m}}{\|\ \mathbf{s}\ \|} + \frac{\mathbf{m}' \wedge \mathbf{s}}{Z} = 0$$

$$\Longleftrightarrow$$

$$-\frac{\mathbf{m}' \wedge \mathbf{s} \wedge \mathbf{F}\ \mathbf{m}}{\|\ \mathbf{s}\ \|} + (\mathbf{m}' \wedge \mathbf{s}) \underbrace{\left(\mathbf{r}^T\mathbf{m} + \frac{\|\ \mathbf{s}\ \|}{Z}\right)}_{\pi_\mathbf{m}} = 0$$

But $\mathbf{m}' \wedge \mathbf{s} \wedge \mathbf{F}\ \mathbf{m} = \underbrace{\left(\mathbf{m}'^T\mathbf{F}\mathbf{m}\right)}_{0}\mathbf{s} - \left(\mathbf{m}'^T\mathbf{s}\right)\mathbf{F}\ \mathbf{m}$, thus following [39] we obtain:

$$\pi_\mathbf{m} = \frac{\|\ \mathbf{F}\ \mathbf{m}\ \|}{\|\ \mathbf{s}\ \|}\left[\frac{\mathbf{s}^T\mathbf{m}'}{\|\ \mathbf{s} \wedge \mathbf{m}'\ \|}\right] \tag{30}$$

We can write this equation because we have shown that the two vectors $(\mathbf{s} \wedge \mathbf{m}')$ and $\mathbf{F}\ \mathbf{m}$ are collinear and are in the same direction. Moreover, we made the assumption that $\mathbf{m}'^T\mathbf{F}\ \mathbf{m} = 0$ thus if we use equation (30) to compute $\pi_\mathbf{m}$ we add the residual error of $\mathbf{m}'^T\ \mathbf{F}\ \mathbf{m}$ to the result.

However, we also have:

$$(\mathbf{m}' \wedge \mathbf{s})^T\ (\mathbf{m}' \wedge \mathbf{s} \wedge \mathbf{F}\ \mathbf{m}) = -\left(\mathbf{m}'^T\mathbf{s}\right)(\mathbf{m}' \wedge \mathbf{s})^T\ \mathbf{F}\ \mathbf{m}$$

and we can compute $\pi_\mathbf{m}$ without any assumption on $\mathbf{F}$ using a more stable equation:

$$\pi_\mathbf{m} = \frac{\left(\mathbf{m}'^T\mathbf{s}\right)\left((\mathbf{s} \wedge \mathbf{m}')^T\ \mathbf{F}\ \mathbf{m}\right)}{\|\ \mathbf{s}\ \|\ \|\ \mathbf{s} \wedge \mathbf{m}'\ \|^2} \tag{31}$$

since this equation is not biased even if $\mathbf{m'}^T\ \mathbf{F}\ \mathbf{m} \neq 0$ due to numerical errors. Thus this is an improvement of the original equation. We note that this equation has ever been used to estimate coplanar structures in [41] but it was not used to perform reconstruction and moreover to perform affine reconstruction.

Thus we reconstruct the depth using the *depth from motion equation*:

$$\delta_{\mathbf{m},\mathbf{H}_\infty} = \frac{\| \mathbf{s} \|}{Z} = \pi_{\mathbf{m}} - \left(\mathbf{r}^T \mathbf{m}\right) \tag{32}$$

3.2 Relating the signs and the scale factors

Given a set of correspondences, we can compute a F-matrix up to an unknown scale factor as if $\mathbf{m'}^T\ \mathbf{F_1}\ \mathbf{m} = 0$, we also have $\mathbf{m'}^T\ \mathbf{F_2}\ \mathbf{m} = 0$ if $\mathbf{F_1} \equiv \mathbf{F_2}$. From this matrix, we compute $\mathbf{s}$ also up to a scale factor, since $\mathbf{s}$ is the null vector of $\mathbf{F}$ as presented in 2.3.2. However, we can relate the sign of $\mathbf{s}$ and of $\mathbf{F}$ using equation (28).

Moreover if we want to reconstruct affine or Euclidean depth, we need $\mathbf{H}_\infty$ which may be computed also up to a scale factor (in the case of a rotation for instance). We can relate here not only the signs of $\mathbf{H}_\infty$ and $\mathbf{F}$ but also the scale factors as $\tilde{\mathbf{s}}\ \mathbf{H}_\infty = \mathbf{F}$.

Thus we can reconstruct the affine depth up to an unique scale factor and up to an additional sign. Therefore we take the absolute value of equation (32) to compute $\delta_{\mathbf{m},\mathbf{H}_\infty}$.

As in the case of the zoom, we note that if we suppose an a-priori on the direction of the vector $\mathbf{s}$, we can thus find $\mathbf{F}$ and $\mathbf{H}_\infty$ with their real signs and compute exactly Z up to a positive scale factor. un-calibratedThis will be extensively used in the sequel.

3.3 Using the zoom displacement for affine calibration

As described previously in 2.3.3 our goal is to use singular displacements to recover affine calibration. Moreover, if we analyze the zoom displacement as given in [39], we can recover affine calibration as described elsewhere.

For this, we must assume that:

- We are able to compute all the collineations in an image,
- some correspondences are at the horizon, enough to have estimated $\mathbf{H}_\infty$, although we don't know which H-matrix is $\mathbf{H}_\infty$.
- We know if we're zooming in or out. We add this information to reconstruct $\mathbf{F}$ and $\mathbf{s}$ with their signs because we can't predict the sign of $\mathbf{s}$ in an case.

We are then able to extract $\mathbf{H}_\infty$ from the set of collineations as we can select extremal planes in the scene as described in [26].But if we can't predict the sign of

s, we may select the furthest fronto-parallel plane but also the nearest one. In this case, where we can't predict the sign of **s** we can only recover affine calibration if we suppose that there is no fronto-parallel plane in front of the scene (see figure 3).

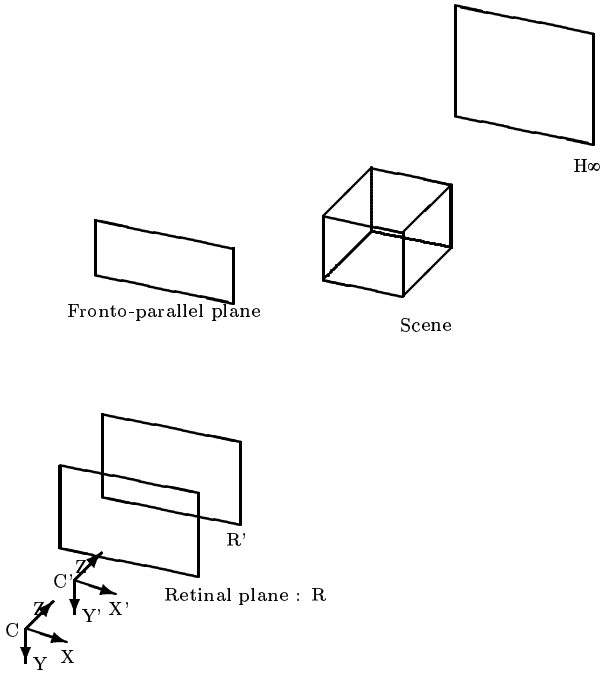

Figure 3: a scene with two extremal fronto-parallel planes

Always in the case of the zoom displacement, we can also reconstruct depth using a fronto-parallel plane. As previously, we have to suppose that we know the sign of **s** to be able to reconstruct the depth and not its opposite.

Moreover we can easily extend this case to the case of a translation and a zoom since we model the zoom with a translation and a modification of the intrinsics parameters.

4 Implementation of a hierarchical estimator

In this chapter we now, make profit of the previous developments to propose a, hopefully complete, taxonomy of all what can be recovered depending on the available calibration.

4.1 What can be really seen without calibration?

4.1.1 Geometric and kinematic constraints: a set of constraints

Let us now enumerate all geometric and kinematic constraints that can be analyzed with the previous mechanism. Thanks to several studies in the field where this problem is developed [9, 45, 25] we only have to review the exhaustive list of constraints. This follows:

- **Projective geometry:**
  - $\mathcal{P}_a$: *Do three points or more belong to a single line ?*
  - $\mathcal{P}_b$: *Do three lines or more intersect on a point ?*
  - $\mathcal{P}_c$: *Are points undergoing the same planar displacement ?*

    Considering our hierarchical model, several other information can be analyzed, as summarized now:
    - $\mathcal{P}_{c.1}$: *Are five non-collinear points or more coplanar(*) ?*
      (*) *considering any collineation non necessarily compatible with a rigid displacement.*
    - $\mathcal{P}_{c.3}$: *Are four non-collinear points or more coplanar(*) ?*
      (*) *assuming their rigid displacement is known.*
    - $\mathcal{P}_{c.4}$: *Are four non-collinear points or more coplanar (*) ?*
      (*) *assuming that their retinal displacement is a retinal displacement.*
    - $\mathcal{P}_{c.5}$: *Are four non-collinear points or more coplanar (*) ?*
      (*) *assuming their 3d-displacement is a pure translation.*
  - $\mathcal{P}_d$: *Are points undergoing the same rigid displacement ?*

    Considering our hierarchical model, several other information can be analyzed, as summarized now:
    - $\mathcal{P}_{d.1}$: *Have eight points or more the same rigid displacement ?*
    - $\mathcal{P}_{d.2}$: *Have three points or more the same motion, assuming it is a pure translation ?*
    - $\mathcal{P}_{d.3}$: *Have four points or more the same motion, assuming it is has no translation ?*
    - $\mathcal{P}_{d.4}$: *Have six points or more the same motion, assuming it is has a fixed axis rotation ?*
    - $\mathcal{P}_{d.5}$: *Have four points or more the same motion, assuming it corresponds to a retinal displacement ?*
    - $\mathcal{P}_{d.6}$: *Have six points or more the same motion, assuming it translation is parallel to the image plane ?*
  - $\mathcal{P}_e$: *Is a planar displacement compatible with a rigid displacement ?*

- $\mathcal{P}_f$: *Are two planar/rigid displacements compatible ?*

- **Affine geometry.** Let us now consider that we have an estimation of $\mathbf{r}$, i.e. of the collineation of the plane at infinity, i.e. an affine calibration.

  - $\mathcal{A}_a$: *Is a point not at the horizon ?*
  - $\mathcal{A}_b$: *Are two or more lines not parallel ?*
  - $\mathcal{A}_c$: *For three collinear points, is M_0 the middle of M_1 and M_2 ?*
  - $\mathcal{A}_d$: *Are two consecutive angles not equal ?*
  - $\mathcal{A}_e$: *Is a point behind another ?*
  - $\mathcal{A}_f$: *Is the rotation axis not parallel to the image plane ?*

- **Euclidean geometry.** Let us now consider that we have an estimation of the intrinsic parameters as obtained from equation (26).

  For the following information:

  - $\mathcal{E}_a$: *Is a line vertical ?*
  - $\mathcal{E}_b$: *Is an angle orthogonal ?*
  - $\mathcal{E}_c$: *What is the value of an angle ?*
  - $\mathcal{E}_d$: *What is the absolute distance of a point (up to global scale factor) ?*
  - $\mathcal{E}_e$: *What is the direction of the translation ?*
  - $\mathcal{E}_f$: *What is the angle and axis of the rotation ?*

  we must use the Euclidean reconstruction, up to a scalar factor $||\mathbf{s}||$. The equations are obvious and will not be rewritten here, although they will be implemented in the last section.

4.1.2 Using random sampling to analyze the data.

The algorithm to estimate these relations is :

- either to be implemented as an interactive process, which answer a "question"; given a set of points, and a geometric relation, the module give a answer on the existence of this relation;

- or to be implemented as a more or less randomized algorithm in which a set of points, and a geometric relation is randomly selected and then tested.

The second option is implemented here, but using random sampling to analyze the data might lead to intractable algorithms.

4.1.3 Hierarchical clustering of points.

The grouping of points in terms of collinear, planar or rigid structures is a particular case of the different geometric of kinematic properties which allows to generate clusters of data. However this is to be done first because we must avoid introduce singular configurations for other estimates.

As discussed before, we can also take into account line-segments represented here by their end points and integrate this data in our system.

4.1.4 Integration along an image sequence

Let us now discuss how we represent the data along an image sequence. We consider an image sequence of N views. The main fact here is the following, *the higher the disparity between two views, the better the numerical stability of the estimate*. Following this remark, we might simply track points along an image sequence and then estimate the displacement between the two extreme views.

Moreover, from the previous equations, it is clear that as soon as the calibration parameters are known for one view, they can be propagated along all image sequence. Similarly, for a point, as soon as its projective proximity, or even Euclidean depth is known in one view, it can be calculated for all other views.

Finally it must be noted that if we want to estimate some particular class of displacements they must correspond to some specific segments of displacements such as fixed axis rotation or a pure translation for which the robot has performed a particular action.

It is clear that if we have some information about the displacement of the projections of points along the image sequence, we can relate the parameters from one view to another. However, we will not introduce this new constraint here because *we do not assume that the potential feature is stationary, whereas it can have any rigid displacement*. Obviously, if the displacement is a general rigid displacement, we definitely cannot relate the different values of.

These remarks lead to integrate the data along an image sequence not using a kind of feed-forward mechanism, one view after another, but to deal globally with the sequence. Anyway, the hierarchical estimator takes point correspondences as input and the estimation of these correspondences is another part of the problem. However, in practice we have two solutions:

- compute correspondences between the first and the last view using a correlation program. The program developed in the Robotvis project, the image-matching program, is based on correlation and robust estimation techniques and allows to estimate correspondences between two views.

- track correspondences in the image sequence and consider only the two extreme positions of the points in the first and last view. An implementation of this solution is described in details in appendix C

4.2 Implementing the hierarchical estimator

Such an estimator is designed to analyze the scene's structure under different calibration levels and under different constraints as described in 4.1.1 and in figure 4.

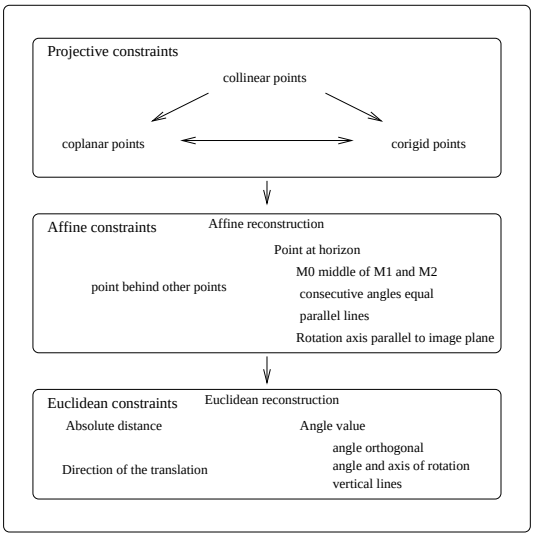

Figure 4: Calibration level and constraints

The algorithm is based on the following hierarchy:

- Using a motion module as described in [39], we estimate if the displacement is a particular displacement.
- We then estimate all collinear points by testing all the possibilities as if n is the number of points the complexity is n^3. We notice that we could use a randomized algorithm to detect these points but in this particular case we can reasonably compute all possibilities. Once detected we can represent all the points on a line segment by the two extremal points and remove all the other points from the list of available points.
- All projective constraints are then analyzed.
- If the particular displacement allows to compute $\mathbf{H}_\infty$, we calibrate the system.
- We then analyze all affine constraints.
- We finally recover Euclidean calibration and compute Euclidean quantities.

This sequential heuristic allows us to avoid having to randomize the estimation of all constraints which would have lead to an intractable algorithmic complexity.

Another part of the problem is that we must deal with several different data types. For instance:

- lines are represented as vectors,
- angles, distances are represented as real numbers,
- collineations are represented as matrices.

All these types are used to represent constraints on the 3D scene and the program has to deal with them. We actually use a object-oriented representation, even if we use the C language for efficiency purpose.

4.2.1 Overview of the algorithm

We apply the following algorithm:

- Select a constraint according to the order defined previously
- Search correspondences satisfying this constraint:
  - Select randomly the point correspondences and all the auxiliary data needed for the constraint estimation
  - Estimate the constraint, return a residual error and a result.
  - According to the previous residual error, keep the result

Of course, since the list of points decreases in length at each step we are sure of the convergence of this algorithm.

We note that the result may have several types but this type is known in function of the constraint. For instance if we interest us to the constraint: "$\mathcal{P}_d$: *Are points undergoing the same rigid displacement ?*", the result will be a fundamental matrix with a list of points corresponding to the displacement.

The program contains severals lists. There is a list for the available constraints which allows to select randomly a constraint according to the order defined previously. When a constraint has been selected once, it's kept out of the list in order to prevent a constraint to be selected several times. Such a list allows to consider calibration as a particular constraint: When calibration is performed and if the calibration can be evaluated (particular displacement), we simply insert all the constraints corresponding to the calibration level in the available constraints list.

Another list is used to represent the available data, that we call primitives. A constraint is defined by the data which are needed to estimate it, by the result type which is produced by the estimation module and by an estimation routine. This representation allows to have a dynamical representation of the scene: a constraint estimator needs data but also produces data which are used by other estimators. When we have chosen a constraint, we initialize a local list

containing the available data which may be selected. Each time we find a set of data satisfying this constraint, we remove these data from the local list in order to improve efficiency.

We remark here that several constraints may produce the same type of data. It is the case for instance for the estimation of a collineation and the estimation of a collineation compatible with a fundamental matrix. The constraint are not the same but they produce the same data : a collineation !

Therefore we have two global lists of data:

- One list contains all the data available of a given type. For instance we will find in this list all the collineations available. This list is used to search data for the estimation module.
- The other list contains all the data satisfying a given constraint, i-e all the results obtained during the estimation of a constraint. This list represents the dynamic representation of the scene.

4.2.2 Estimation routines and optimization

We need several estimation routines to implement our modules: least-square, generalized (non linear) least-squares, robust (least-median squares) estimators, etc ... Since we reuse well known mechanisms in the field we will not review these methods here. We:

- Estimate a quantity given a set of point correspondences and additional data. The 'quantity' is defined by the constraint. Additional data are all the data different from the correspondences that the estimation function needs. For instance, if we try to compute a collineation compatible with a fundamental matrix, the estimation function needs not only the point correspondences but also a fundamental matrix.
- If the residual error is smaller than a threshold, create a list of correspondences compatible with the estimated result. If a robust estimation was performed, eliminate all the outliers.
- For each point correspondence, compute if it is compatible with the previous result. In this case, add this data to the list.

To be as efficient as possible when estimating collineations and fundamental matrices (see appendix A), the estimation is computed in two stages:

- A first estimation on a quadratic criterion allows to eliminate outliers by using a robust estimation method.
- Then another estimation is performed to refine the previous result. This paradigm might be improved using recent developments.

We compute several estimation at each step with different constraints on the displacement. The estimation having the smaller residual error will be taken into account. This method allows to find better parameterization as presented in 2.3.7.

We note here that the estimator should not only estimate collineations or fundamental matrices but also quantities easier to estimate. For instance, to calculate a line passing throw the points $\mathbf{a}$ and $\mathbf{b}$, we can either minimize a criterion as presented in appendix A or only compute quickly $\mathbf{l} = \mathbf{a} \wedge \mathbf{b}$. Thus we can also estimate if the point $\mathbf{c}$ is on the line by checking if $\mathbf{c}^T\ \mathbf{l} = 0$.

4.2.3 Constraint management in the algorithm.

This is also an important part of the algorithm. In fact, this procedure consists not only in an insertion into a list. We have to:

- check if this result has not been yet inserted,
- check if the points corresponding to this result are not yet in another list of points belonging to a previous result of the same constraint,
- check if we have to merge this result with a previous result,
- choose a few points to represent all the others. For instance, if we find collinear points, we'll represent this set of points using the extremities of the segments. A very simple method to find quickly the two extremities is to perform a quick sort on the x axis.

The most important part of this module is the merging function, which allow to reduce the size of the database.

5 Experimental results

5.1 Synthetic data

5.1.1 Generation of the data

We generate the synthetic data, using a set of three dimension data points, and project these points on a virtual camera, simulating a displacement of the camera. Thus we generate a file containing the projected points for the first and for the second projection. We use several different scenes to test the algorithm:

- A scene containing a synthetic grid, which allows to verify that the collinear points are found and to perform affine and Euclidean reconstruction.

- A scene containing several planes, which allows to analyze the behavior of the algorithm when the scene contains more than two planes and when the scene consists not only in a calibration grid. To generate this scene, we randomly select points on a few planes, and consider points which are not corresponding to geometric particular points. We obtain the scene in figure 5. We use two scenes of such: one containing 3 planes and the other

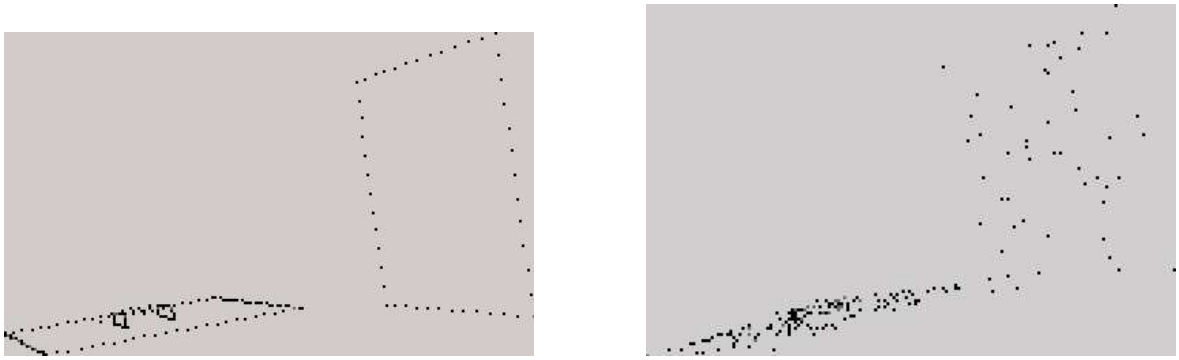

Figure 5: A scene containing 5 planes and the data selected randomly on the planes

containing 5 planes.

- finally, a scene containing a synthetic grid and some fronto-parallel planes, which allows to reconstruct affine depth using a fronto-parallel plane.

We choose the following projection matrix for the virtual camera:

$$\mathbf{A} = \begin{pmatrix} 800 & 0 & 256 \\ 0 & 800 & 256 \\ 0 & 0 & 1 \end{pmatrix}$$

This projection matrix corresponds indeed to a camera with a focal length $f = 800$ and a principal point $(u_0, v_0) = (256, 256)$.

5.1.2 Projective constraints and calibration

Collinear points. We use the synthetic grid (see figure 6) as it contains many coplanar points. The displacement is a pure translation on the x axis with $\| t \| = 10$. The grid translation from the principal point is $\mathbf{t} = \begin{pmatrix} 500 & 500 & 5000 \end{pmatrix}^{\mathbf{T}}$. As presented in figure 7, we have detected the principal lines in the scene. We note here that the threshold has an important influence and that some of the points are not collinear due to numerical approximations when computing the projection and the displacement.

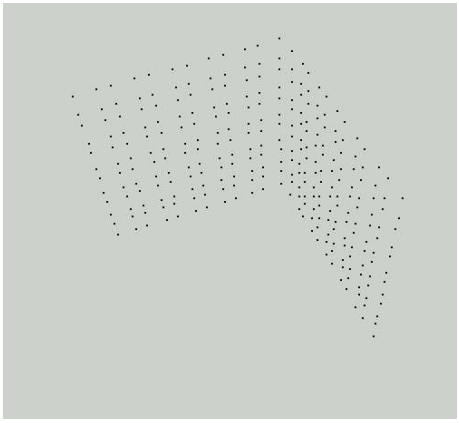

Figure 6: The synthetic grid

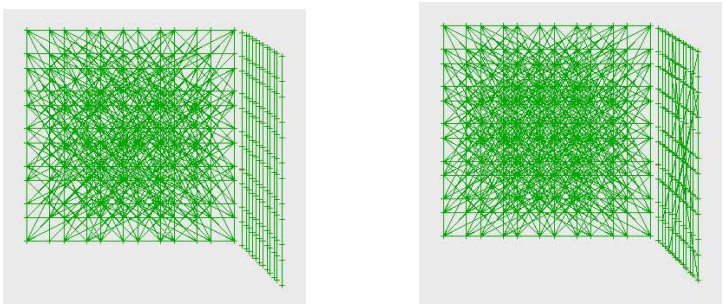

Figure 7: Estimation of collinear points with different threshold : $1e-11$ and $1e-7$

Co-rigid points: translations. With the same scene and the same displacement, we now detect the co-rigid points. The synthetic grid contains 288 point correspondences. We obtain the following fundamental matrix:

$$\mathbf{F} = \begin{pmatrix} 0 & -0 & 0 \\ 0 & 0 & -0.707107 \\ -0 & 0.707107 & 0 \end{pmatrix}$$

The pure retinal translation parameterization was chosen automatically by the algorithm to compute this result. Using this matrix we can perform projective calibration and we compute $\mathbf{s} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T$. We easily verify that the estimated matrix corresponds to a pure translation along the x axis. In this case the fundamental matrix has the following form:

$$\mathbf{F} = \begin{pmatrix} 0 & -s^2\frac{f'}{f} & \frac{-s^2\left(fv'_0 - f'v_0\right)+s^1 f}{f} \\ s^2\frac{f'}{f} & 0 & \frac{s^2\left(fu'_0 - f'u_0\right)-s^0 f}{f} \\ -s^1\frac{f'}{f} & s^0\frac{f'}{f} & -\frac{s^1\left(fu'_0 - f'u_0\right)+s^0\left(f'v_0 - fv'_0\right)}{f} \end{pmatrix}$$

and the epipole:

$$\mathbf{s} = \begin{pmatrix} f't^0 + u'_0 t^2 & f't^1 + v'_0 t^2 & t^2 \end{pmatrix}$$

If we change the displacement and add a translation on the y axis, the translation is now $\mathbf{t} = \begin{pmatrix} 100 & 200 & 0 \end{pmatrix}^{\mathbf{T}}$ and we obtain:

$$\mathbf{F} = \begin{pmatrix} 0 & -0 & 0.632456 \\ 0 & 0 & -0.316228 \\ -0.632456 & 0.316228 & 0 \end{pmatrix}$$

and:

$$\mathbf{s} = \begin{pmatrix} 0.447214 & 0.894427 & 0 \end{pmatrix}$$

The parameterization chosen corresponds again to the pure retinal translation constraint. We perform a last translation with $\mathbf{t} = \begin{pmatrix} 100 & 50 & 100 \end{pmatrix}^{\mathbf{T}}$. This translation does not correspond to a retinal translation anymore as previously studied, but to a general translation. We obtain:

$$\mathbf{F} = \begin{pmatrix} -0 & -0.000568793 & 0.373128 \\ 0.000568793 & -0 & -0.600645 \\ -0.373128 & 0.600645 & -0 \end{pmatrix}$$

and:

$$\mathbf{s} = \begin{pmatrix} -0.849441 & -0.527683 & -0.000804 \end{pmatrix}$$

Co-rigid points: zoom displacement. We simulate a zoom displacement with a translation $\mathbf{t} = \begin{pmatrix} 10 & 0 & 100 \end{pmatrix}^{\mathbf{T}}$ and a variation of the focal length $f' = f + 200$. We obtain the following results for all the correspondences:

$$\mathbf{F} = \begin{pmatrix} 0 & -0.00198657 & 0.508561 \\ 0.00198657 & 0 & -0.491286 \\ -0.508561 & 0.491286 & 0 \end{pmatrix}$$

$$\mathbf{s} = \begin{pmatrix} -0.694784 & -0.719213 & -0.00280943 \end{pmatrix}$$

We note that the algorithm has chosen the zoom constraint as expected.

Co-rigid points: retinal displacement We can also study the case of a translation and a rotation using another scene. We use the scene containing 5 planes and constructed randomly (figure 5). To simplify the results, we only focus on a translation on the x axis ($t^0 = 100$) and a rotation on the z axis ($\theta = 0.2$). The fundamental matrix is now:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -sin(\theta) \\ s^0 sin(\theta) & s^0 cos(\theta) & -s^0(u_0 sin(\theta) + v_0 cos(\theta) - v_0) \end{pmatrix}$$

The displacement estimated is the retinal displacement and we find for all the point correspondences:

$$\mathbf{F} = \begin{pmatrix} 7.11968e-11 & 1.68395e-10 & -8.07741e-08 \\ -1.59771e-10 & 5.24371e-11 & 0.0125792 \\ -0.00489854 & -0.0115863 & 0.999842 \end{pmatrix} \quad \mathbf{s} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

Another advantage of this algorithm is that we can deal with several displacements. We take a scene a simulate two different moves in this scene. As seen in 2.2.6, we can distinguish between these two displacements if they don't correspond to the same translation (up to a scale factor). We apply the following translations to the two objects of the scene:

$$\mathbf{t_0} = \begin{pmatrix} 0 & 0 & 500 \end{pmatrix}^T$$

and:

$$\mathbf{t_1} = \begin{pmatrix} 100 & 100 & 0 \end{pmatrix}^T$$

And we obtain the following results:

$$\mathbf{F_0} = \begin{pmatrix} 0 & 0.00195312 & -0.499998 \\ -0.00195312 & 0 & 0.499998 \\ 0.499998 & -0.499998 & 0 \end{pmatrix}$$

$$\mathbf{F_1} = \begin{pmatrix} 0 & -0 & 0.5 \\ 0 & 0 & -0.5 \\ -0.5 & 0.5 & 0 \end{pmatrix}$$

Fair enough, these results correspond to the theoretical values up to a scale factor.

Coplanar points. We consider again the grid scene and simulate a pure translation on the x axis and on the y axis: $\mathbf{t} = \begin{pmatrix} 100 & 100 & 0 \end{pmatrix}^{\mathbf{T}}$. The grid is translated from the principal point with $\mathbf{t_0} = \begin{pmatrix} 500 & 500 & 5000 \end{pmatrix}^{\mathbf{T}}$. The camera is looking at the first plane of the grid such as its normal is on the z axis of the camera. We estimate two collineations and all the correspondences belong to one of these collineations. We represent the planar structure using a Delaunay triangulation performed on the set of points given by the algorithm (see figure 8). The triangulation is not used for the estimation but only for graphics.

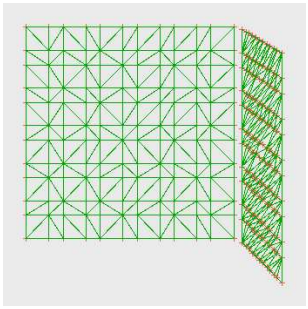

Figure 8: The synthetic grid : estimation of planar structures

The theorical form for a collineation under a pure translation is:

$$\mathbf{H} = \begin{pmatrix} 1 + \frac{s^0 n^0}{fd} & \frac{s^0 n^1}{fd} & -\frac{s^0(u_0 n^0 + v_0 n^1 - n^2 f)}{fd} \\ \frac{s^1 n^0}{fd} & 1 + \frac{s^1 n^1}{fd} & -\frac{s^1(u_0 n^0 + v_0 n^1 - n^2 f)}{fd} \\ \frac{s^2 n^0}{fd} & \frac{s^2 n^1}{fd} & 1 - \frac{s^2(u_0 n^0 + v_0 n^1 - n^2 f)}{fd} \end{pmatrix} \tag{33}$$

where $\mathbf{n}$ is the plane normal and d its distance to the origin. Here, we find a first collineation, corresponding to the fronto-parallel structure:

$$\mathbf{H_1} = \begin{pmatrix} 0.0440653 & -2.02393e-17 & 0.705044 \\ 0 & 0.0440653 & 0.705044 \\ 0 & 0 & 0.0440653 \end{pmatrix}$$

We can easily check in this case (translation on the x and on the y axis) that the normal of this plane is on the z axis since $H^{0,1}$, and $H^{1,0}$ are null but $H^{0,2}$ is not null. Again the result is obtained up to a scale factor.

The second collineation is:

$$\mathbf{H_2} = \begin{pmatrix} 0.0268364 & -1.25964e-10 & -0.706439 \\ 0.00275953 & 0.0240769 & -0.706439 \\ 0 & 0 & 0.0240769 \end{pmatrix}$$

We also verify that the normal of this plane is on the x axis since $H^{1,0}$ is not null as expected.

We can perform another translation only on the x axis ($t^0 = 100$). In this case we obtain these two matrices:

$$\mathbf{H_1} = \begin{pmatrix} 0.062137 & 0 & 0.994192 \\ 0 & 0.062137 & 0 \\ 0 & 0 & 0.062137 \end{pmatrix}$$

$$\mathbf{H_2} = \begin{pmatrix} 0.037917 & 3.22278e-10 & -0.998122 \\ -0 & 0.0340181 & -0 \\ 0 & 0 & 0.0340181 \end{pmatrix}$$

In this case, as $H_2^{0,0}$ is different (and greater) from $H_2^{1,1} = H_2^{2,2}$, the normal of the second plane is on the x axis. More over as $H_1^{0,2}$ is not null and that $H_1^{1,0} = 0$ and $H_1^{0,0} = H_1^{1,1} = H_1^{2,2}$, the normal of the first plane is on the z axis.

In the case where the normal is on the z axis, we have $H_1^{1,2} \equiv \frac{s^0}{d}$. To verify our results, we just have to note that we perform a pure translation. In this case we should have $H_1^{0,0} = H_1^{1,1} = 1$ as $n^0 = n^1 = 0$, which is the case. We can multiply our result by $\frac{1}{H_1^{1,1}}$ and recover the scale factor **on the collineation** in this particular case. We obtain finally:

$$\mathbf{H_2} = \begin{pmatrix} 1.000000000 & 0 & 16.00000000 \\ 0 & 1.000000000 & 0 \\ 0 & 0 & 1.000000000 \end{pmatrix}$$

We can deduce that $\frac{s^0}{d} = 16.00$. As $\mathbf{s} = \mathbf{A't} = \begin{pmatrix} 80000 & 0 & 0 \end{pmatrix}^T$, we can calculate $d = 5000$. We must note here that in pratice **we can't compute s**. If we have calculated it, it is to verify our results.

Thus we can check that d equals the distance where the grid stand from the optical center as we had translated it before the simulation of the displacement in $\mathbf{t_0} = \begin{pmatrix} 500 & 500 & 5000 \end{pmatrix}^{\mathbf{T}}$.

Let use take a scene containing 3 orthogonal planar structures(9) with the correspondences randomely selected. We choose the orientation of the camera such as having the normal of plane 1 on the optical axis(z axis).

In the case of a translation ($t^1 = 20$), we compute:

- The collineation corresponding to plane 1

$$\mathbf{H_1} = \begin{pmatrix} 0.23043 & -2.95555e-17 & -3.7831e-15 \\ 9.1271e-07 & 0.23043 & 0.9169 \\ 0 & 0 & 0.23043 \end{pmatrix}$$

  This plane as its normal on the z axis. Moreover, we can check that we have $H^{1,0} = 0$ and $H^{0,0} = H^{1,1} = H^{2,2}$ according to equation (33).

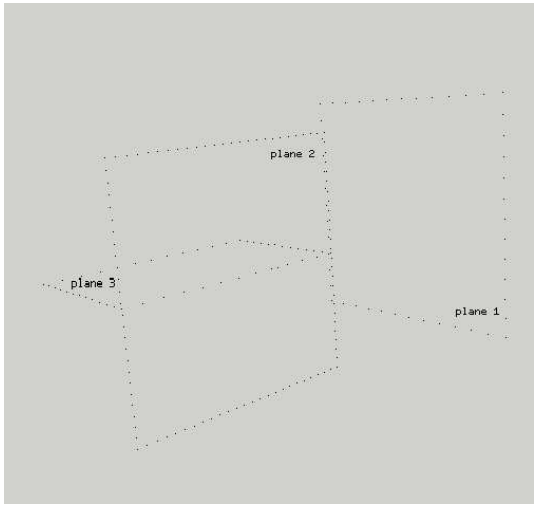

Figure 9: Three othogonal planar structures

- The collineation corresponding to plane 2

$$\mathbf{H_2} = \begin{pmatrix} 0.0388799 & -0 & -7.54509e-16 \\ -0.00388768 & 0.0388799 & 0.997722 \\ 0 & 0 & 0.0388799 \end{pmatrix}$$

  The normal of this plane is clearly on the x axis as $H^{1,0}$ is not null.

- The collineation corresponding to plane 3

$$\mathbf{H_3} = \begin{pmatrix} 0.184885 & -0 & -0 \\ 5.66204e-11 & 0.188583 & -0.946611 \\ 0 & 0 & 0.184885 \end{pmatrix}$$

  This plane has its normal on the y axis since we have $H^{1,1}$ greater than $H^{1,1} = H^{2,2}$ and $H^{1,0} = 0$.

We can verify that in the case of a rotation, each correspondence in the scene is predicted by a unique collineation. Let us consider to the case of a pure rotation around the z axis. As expected, we obtain one collineation for all the correspondences that we can multiply by a scale factor (H) to compare with the theorical value (H_{Th}):

$$\mathbf{H} = \begin{pmatrix} .980066 & -.198669 & 55.962304 \\ .198669 & .980066 & -45.75639 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{H_{Th}} = \begin{pmatrix} .980064 & -.19866 & 55.962125 \\ .198668 & .980064 & -45.756265 \\ 0 & 0 & 1.000 \end{pmatrix}$$

In order to analyze collineations under general displacements and not only singular displacements, we consider a translation on the x and on the y axis and

a rotation around the x axis. However we suppose once more that the calibration parameters are constant as we are going to study the case of zoom displacements later on (see 5.1.4). We obtain the following results, with a general displacement parameterization:

$$\mathbf{H_1} = \begin{pmatrix} 0.00639 & 4.066e-04 & 0.1190 \\ 1.273e-07 & 0.00667 & -0.9928 \\ 7.548e-14 & 1.588e-06 & 0.00586 \end{pmatrix}$$

$$\mathbf{H_2} = \begin{pmatrix} -1.56e-10 & 0.000265 & 0.980691 \\ -0.00208429 & 0.004350 & -0.195469 \\ -8.68e-12 & 1.035e-06 & 0.003820 \end{pmatrix}$$

$$\mathbf{H_3} = \begin{pmatrix} -1.56e-10 & 0.000265 & 0.980691 \\ -0.002084 & 0.004350 & -0.195469 \\ -8.68e-12 & 1.035e-06 & 0.003820 \end{pmatrix}$$

As $\mathbf{H_2}$ corresponds to a plane with its normal on the x axis, we can compare with the theorical collineation of a plane with such a normal under this displacement at a distance d_2 from the optical center:

$$\mathbf{H_{Th2}} = \begin{pmatrix} 1 & 0.063557 + \frac{100}{d_2} & -21.37794 - \frac{25600}{d_2} \\ 0 & 1.043640 + \frac{50}{d_2} & -175.21045 - \frac{12800}{d_2} \\ 0 & .000248 & .91649 \end{pmatrix}$$

Thus we compute d and find $d_2 = -99.999$. We can easily check that this distance corresponds to the distance from the plane to the optical center. We also note that such a plane is projected on a line in the retina. Using the same method we can compute d_1 using the theorical matrix for a plane with its normal on the z axis:

$$\mathbf{H_{Th1}} = \begin{pmatrix} 1 & 0.063557 & -21.37794 - \frac{80000}{d_1} \\ 0 & 1.043640 & -175.21045 - \frac{40000}{d_1} \\ 0 & .000248 & .91649 \end{pmatrix}$$

We find $d_1 = 1999.89$ and the plane is at a distance $d1 = 2000$ from the optical axis. We can also check our results for H_3 and find $d_3 = 499.9978$ "instead" of 500.

5.1.3 Affine reconstruction: pure translation

We perform a pure translation on the x axis . We obtain the following fundamental matrix:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & 3.27072e-16 \\ 0 & 0 & -0.707107 \\ 5.45858e-16 & 0.707107 & 0 \end{pmatrix}$$

Figure 10: Projective and affine reconstruction

Therefore we can recover the affine calibration since $\mathbf{H}_\infty = \mathbf{I}$and compare projective and affine reconstruction : as we know the projection matrix [5], we can estimate the affine transform between our reconstruction and the Euclidean reconstruction ($\mathbf{A^{-1}}$) in order to verify our results:

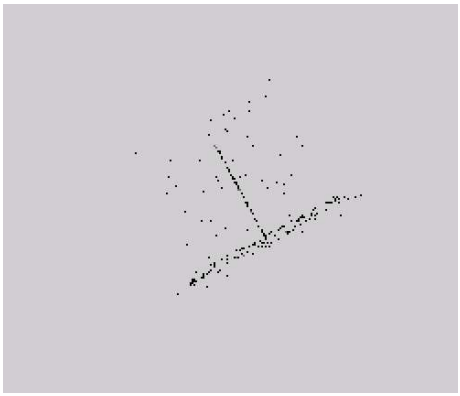

Figure 11: Transformation of the affine reconstruction

We can check our result easily using equation ((20).

We can see that the middle of pairs of points or the median of lines are preserved in the case of the affine reconstruction as the segment corresponding to the vertical plane cut the segment corresponding to the horizontal plane through its middle. Moreover we see that the angles are not preserved as these segment are orthogonal in the scene. This is expected since affine reconstruction preserve the middle and parallel lines but not the angles. Furthermore, we can observe for the euclidean reconstruction that the angles are preserved.

[5]We note that in pratice we can not estimate this transform because we can not estimate the projection matrix.

We obtain similar results using the grid scene under a pure translation $\mathbf{t} = \begin{pmatrix} 100 & 100 & 0 \end{pmatrix}^{\mathbf{T}}$:

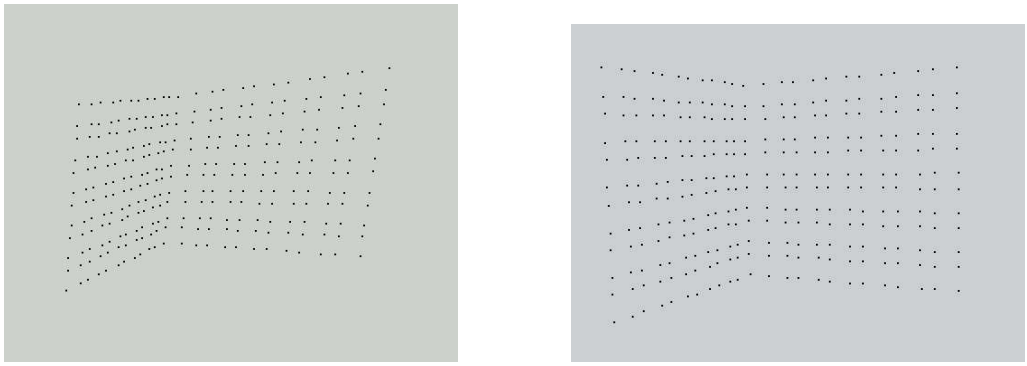

Figure 12: Affine and euclidean reconstruction

5.1.4 The case of a zoom displacement

As described previously, we model a zoom displacement with a translation and a variation of the intrinsic parameters:

$$\mathbf{t} = \begin{pmatrix} 20 & 0 & 200 \end{pmatrix}^T \text{ and } f' = f + 200$$

The scene is translated from the optical center with:

$$\mathbf{t_0} = \begin{pmatrix} 1000 & 1000 & 6000 \end{pmatrix}^T$$

In order to reconstruct with several planes, we use a synthetic scene containing five planes: a calibration grid and another plane between two other fronto-parallel planes. There are three fronto-parallel planes in this scene. As described elsewhere, we are able to distinguish all fronto-parallel plane's collineations (see figure 13) from generic ones.

Moreover if we suppose that we can obtain the sign of the displacement (from odometric cues), the sign of $\mathbf{s}$, we can reconstruct affine depth using the farthest plane as the plane at infinity (see figure 14). Of course, we must not try to reconstruct the depth of points which are on this particular plane !

Once more, the middle is preserved but not the angles as we have an affine reconstruction. Rather than estimating the depth which is known as being an unstable quantity, we can compute the affine retinal proximity $\delta_{m,\mathbf{H}_\infty}$ (the inverse of the depth). This allows us to compute the retinal proximity for all the plane in the scene including the "horizon" [6] as presented in figure 15.

[6] The plane chosen as $\mathbf{H}_\infty$ is considered as being the horizon in the scene. Its retinal proximity is therefore zero and its depth infinite.

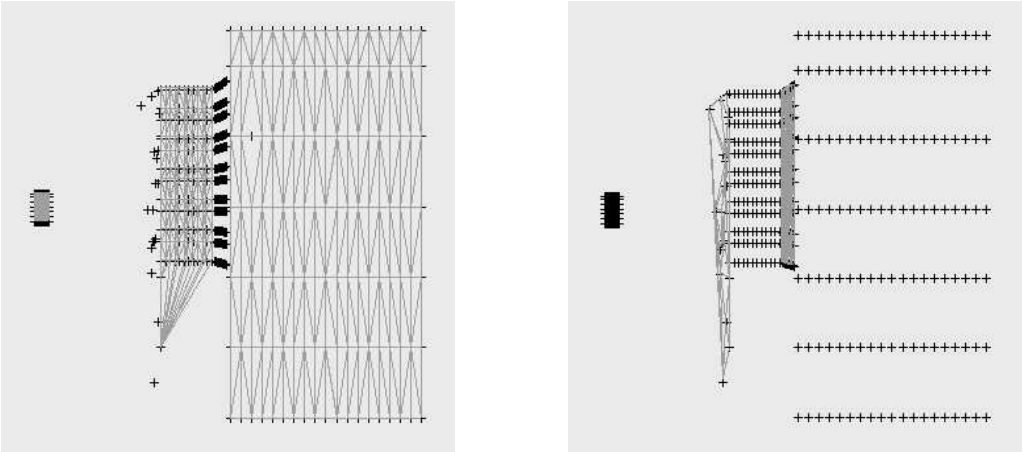

Figure 13: Selection of fronto-parallel planes

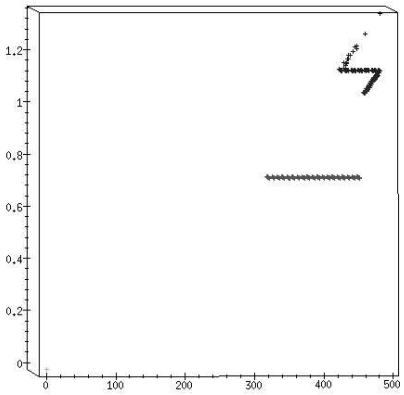

Figure 14: Affine reconstruction (zoom displacement)

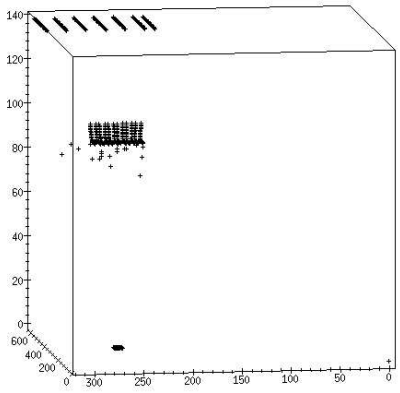

Figure 15: Affine retinal proximity

We observe here that parallel lines are also preserved as expected again.

Using the same apriori, we can also compute the affine retinal proximity using the others fronto-parallel planes. In fact, we estimate a strictely increasing function of the retinal proximity as predicted by our theorical equations.

We obtain the results presented in figure 16 using the last two fronto-parallel planes collineation as $\mathbf{H}_\infty$.

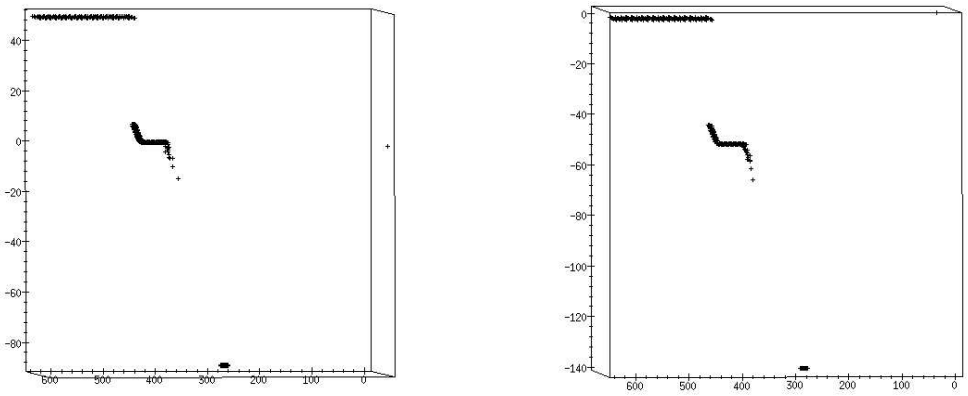

Figure 16: Reconstruction using fronto-parallel planes

Let us now change the displacement:

$$\mathbf{t} = \begin{pmatrix} 2 & 0 & -20 \end{pmatrix}^T \quad \text{and} \quad f' = f - 200$$

If we reconstruct the affine retinal proximity, updating the apriori information on the sign of $\mathbf{s}$, we obtain the results of figure 17.

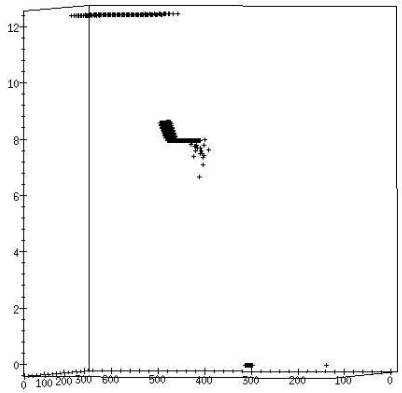

Figure 17: Affine retinal proximity when zooming out

Let us finally take another scene containing three planes and where the correspondences are randomly selected.

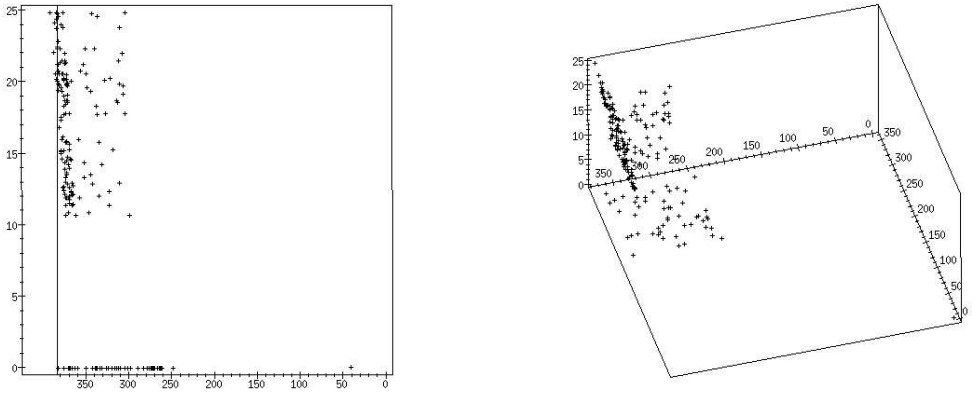

Figure 18: Affine retinal proximity

We can distinguish in the reconstruction (figure 18) the three planes:

- The first plane has a zero affine retinal proximity and corresponds to the plane chosen for the reconstruction.
- We distinguish the second plane on the second figure as it appears as a diagonal plane.

- Then the last plane plane appears as an horizontal plane in the second figure.

We again recover the expected structure of the scene.

5.1.5 Euclidean calibration

Let us perform a pure rotation of x axis on the scene containing the three planes. The angle of the rotation is $\theta = 0.2$. We estimate one collineation for the three planes. We can compare this collineation with the theorical value ($\mathbf{H_{Th}}$):

$$\mathbf{H} = \begin{pmatrix} 1.0 & 0.063574 & -21.377935 \\ 1.182 \times 10^{-8} & 1.043640 & -175.210497 \\ 1.110 \times 10^{-11} & 0.0002483 & 0.916492 \end{pmatrix}$$

$$\mathbf{H_{Th}} = \begin{pmatrix} 1 & 0.063574 & -21.377947 \\ 0 & 1.043640 & -175.210456 \\ 0 & 0.000248 & 0.916492 \end{pmatrix}$$

From $\mathbf{H_{Th}}$ we can compte the matrix O using equation (25):

$$\mathbf{O} = \begin{pmatrix} -6.80 \times 10^{-7} & 0.063574 & -16.274969 \\ 2.106 \times 10^{-9} & 0.063573 & -175.210480 \\ 1.120 \times 10^{-11} & 0.0002483 & -0.0635738 \end{pmatrix}$$

We can compute f, u_0 and v_0 usings equations (26), estimate the projection matrix and compare it to the theorical value:

$$\mathbf{A} = \begin{pmatrix} 800.0008 & 0 & 256.0006 \\ 0 & 800.0008 & 255.9993 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A_{Th}} = \begin{pmatrix} 800 & 0 & 256 \\ 0 & 800 & 256 \\ 0 & 0 & 1 \end{pmatrix}$$

Moreover we estimate the axis of the rotation:

$$\rho = \begin{pmatrix} 1.0000 & .11757e^{-5} & .41458e^{-9} \end{pmatrix}^T$$

Thus we are able, in the case of singular displacement to recover Euclidean calibration.

5.1.6 Adding noise

We add a noise on each match, with a standart deviation of $\sigma = 0.3$. We notice that such a noise corresponds to usual precision of corner detectors. We take the grid scene and perform a translation:

$$\mathbf{t} = \begin{pmatrix} 100 & 0 & 0 \end{pmatrix}^T$$

We obtain an expression of the fundamental matrix that we can compare the theorical value ($\mathbf{F_{Th}}$):

$$\mathbf{F} = \begin{pmatrix} 0 & -3.503e-05 & 0.00623 \\ 3.503e-05 & 0 & -0.7070 \\ -0.00623 & 0.7070 & 0 \end{pmatrix}$$

and:

$$\mathbf{F_{Th}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -0.707107 \\ 0 & 0.707107 & 0 \end{pmatrix}$$

So that we still obtain coherent results. We have not pushed this analysis because a complete set of experimental results is available in [39].

5.2 Real data

5.2.1 Detection of collinear points

Let us take an image sequence of a grid using a robotic system which performs pure translations along the camera's x axis .

Since we want a "pure" translation, we switch off the auto-focus system and set the zoom to a fixed value. The focal length is fixed as $f = 498$ mm and the zoom is set to $z = 596mm$.

We perform the following displacement:

image number	T_x (mm)
1	570
2	560
3	550
4	540
5	530
6	500

We compute correspondences between the first and the last image (70 millimeters of disparity) and find 188 correspondences. If we try to estimate collinear points (in the image's plane), we obtain with a first thresold $S = 1$, we obtain the results shown in figure 19. We notice that we can detect a few horizontal and vertical bi-dimentional lines [7] but also several diagonal lines. In fact if we perform a zoom we can notice (see figure 20 that the points detected are not as well aligned on horizontal and vertical lines than on certain diagonal lines.

The algorithm has selected collinear points in the images and eliminate these points to compute fundamental matrices and collineations.

[7]Note that if we want to estimate collinear space points, we must consider lines different from the epipole as described in appendix A.4 and analyzed in [41]

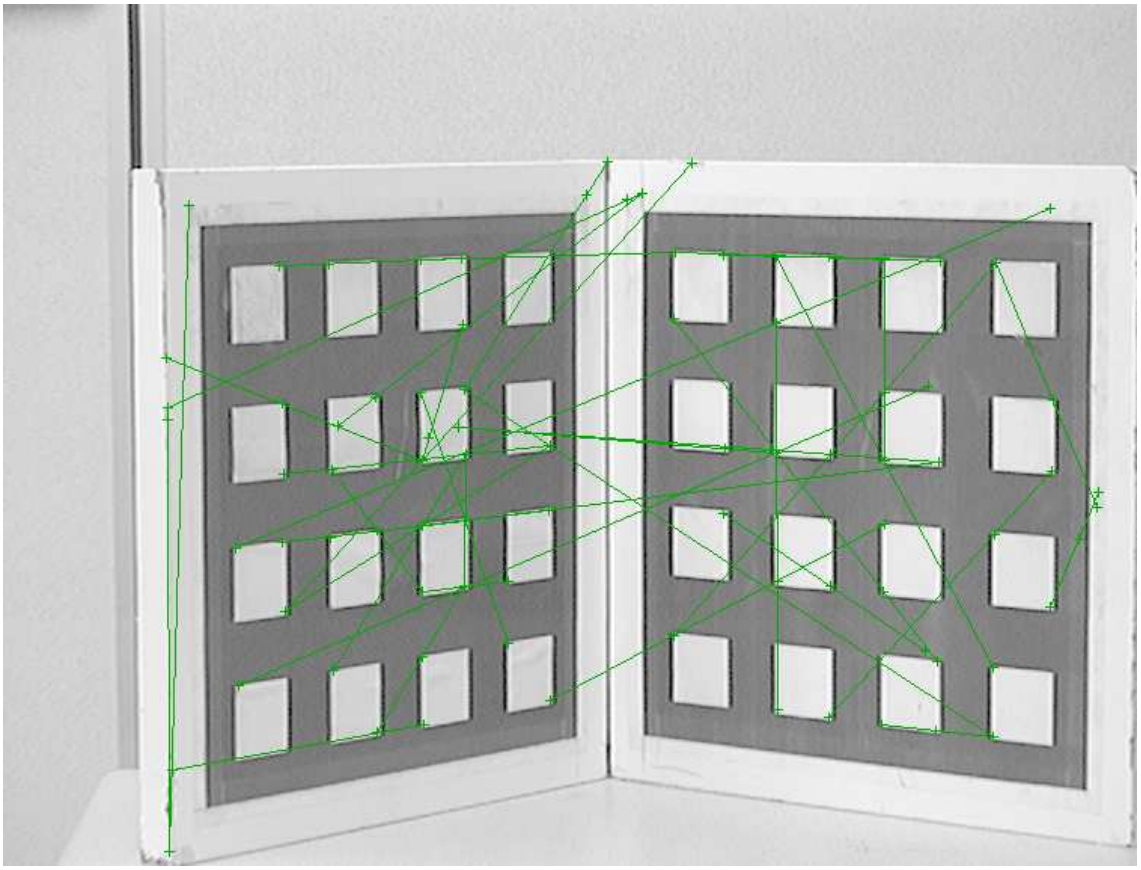

Figure 19: Picking collinear points

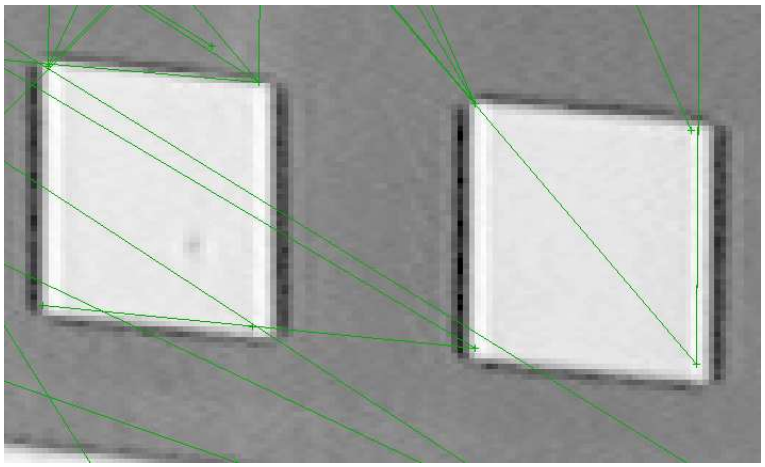

Figure 20: Horizontal points are not strictely collinear

If we modify this thresold, we obtain results presented in figure 21 for a threshold $S = 5$ and for a thresold $S = 10$. We notice that we have estimated more vertical and horizontal lines than for a lower threshold.

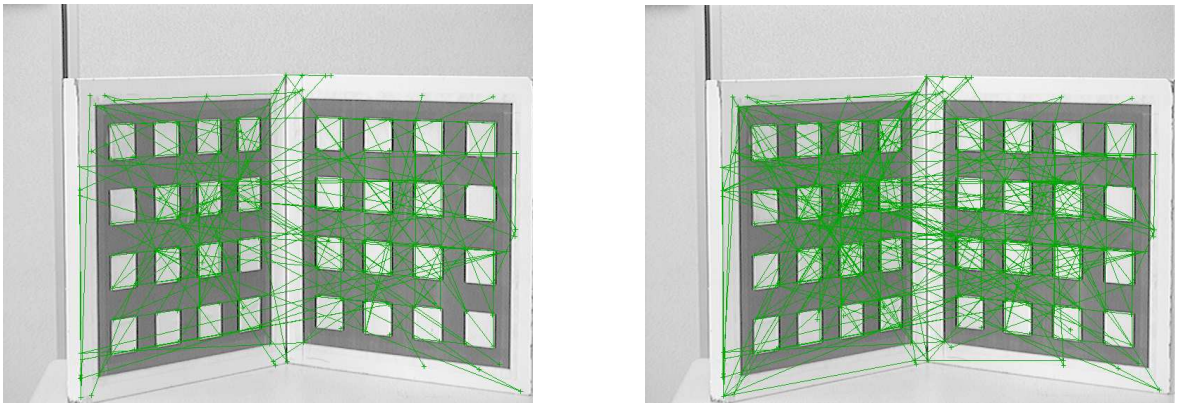

Figure 21: Collinear points for different threshold (5 and 10)

5.3 Corigid and coplanar points on a grid

Using the previous correspondences, we can compute the fundamental matrix. We find the following fundamental matrix (for 184 correspondences):

$$\mathbf{F_{Th}} = \begin{pmatrix} 0 & 5.21447e-06 & -0.00173979 \\ 5.21447e-06 & 0 & -0.707105 \\ 0.00173979 & 0.707105 & 0 \end{pmatrix}$$

and the theorical matrix is:

$$\mathbf{F_{Th}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -0.707107 \\ 0 & 0.707107 & 0 \end{pmatrix}$$

We can also compute two collineations:

$$\mathbf{H_1} = \begin{pmatrix} 0.00968634.47558e-05 & -0.99984 \\ 9.1752e-060.009321380.0072686 \\ 000.00934815 \end{pmatrix}$$

and

$$\mathbf{H_2} = \begin{pmatrix} 0.01215835.97929e-06 & -0.999605 \\ 1.56161e-050.0124929 & -0.0181814 \\ 000.0124694 \end{pmatrix}$$

There are 60 points on the first plane and 54 for the second one (after 10000 steps). We notice that the normals have a component on the x axis (and no component on the y axis) for the two planes and in opposite directions since:

- $H_1^{1,1}$ is greater than $H_1^{2,2}$ and $H_1^{3,3} = H_1^{2,2}$,
- $H_2^{1,1}$ is lower than $H_2^{2,2}$ and $H_2^{3,3} = H_2^{2,2}$.

However we can't estimate if these planes have a component along the z axis because the translation was only performed on the x axis and with such a translation we can't recover $s^0 n^2$ (see 33). Moreover, we can represent these planes using Delaunay triangulation. (see figure 22). Collinear points remain in the scene since

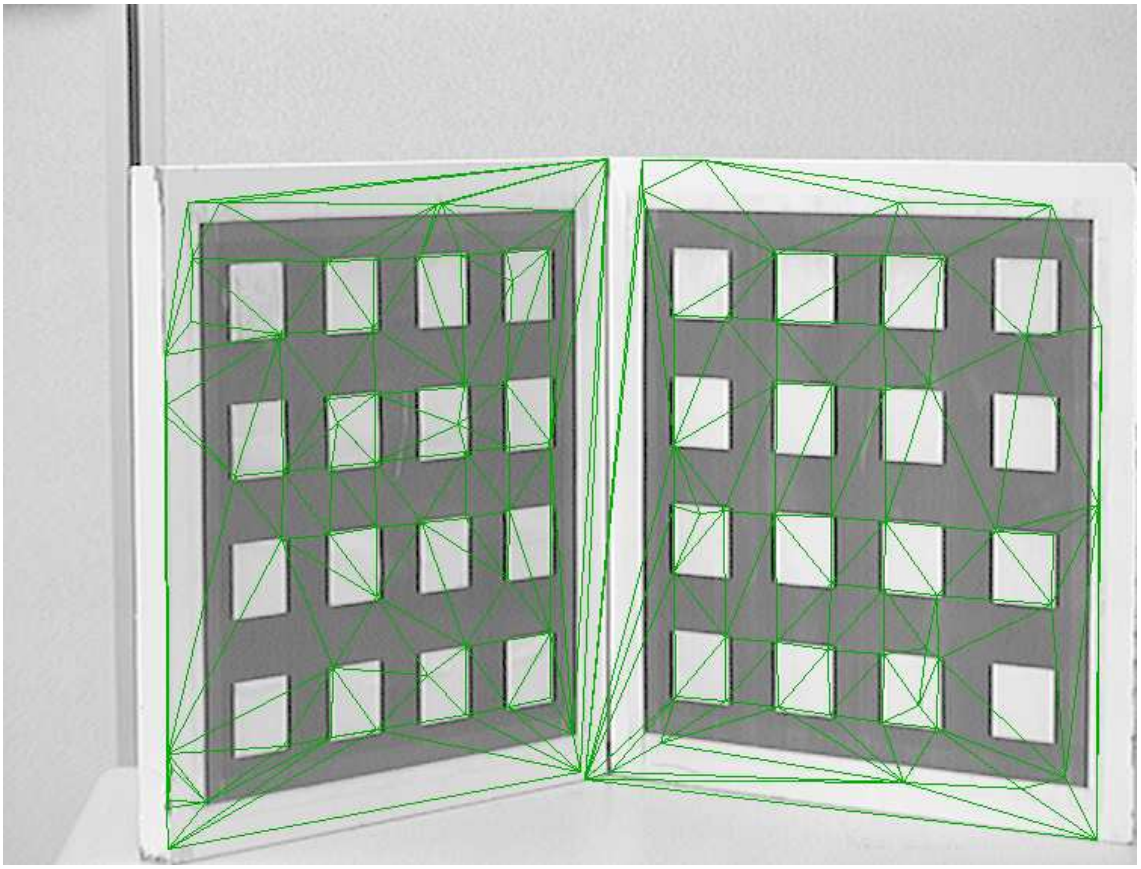

Figure 22: Estimation of planar structures

these collineations have been estimated using the "co-planar" constraint (and not the coplanar and compatible with a fundamental matrix).

5.4 Corigid and coplanar points on an indoors scene

Let us take another scene with three planes. We fix the focal length and the zoom:

$$f = 782 \text{ mm and } z = 0$$

and perform a translation along the x axis (from 590 to 500mm). We obtain a fundamental matrix:

$$\mathbf{F} = \begin{pmatrix} 0 & 4.76495e-06 & -0.00220561 \\ -4.76495e-06 & 0 & -0.707103 \\ 0.00220561 & 0.707103 & 0 \end{pmatrix}$$

If we set the number of iteration to the same number as previously (10000), we obtain only the two main collineations (see figure 23):

$$\mathbf{H_1} = \begin{pmatrix} 0.00764348 & 3.92544e-05 & -0.99992 \\ -4.9159e-06 & 0.00709485 & -0.000297399 \\ 0 & 0 & 0.00708294 \end{pmatrix}$$

and:

$$\mathbf{H_2} = \begin{pmatrix} 0.013372 & 3.25211e-05 & -0.999683 \\ -1.72255e-06 & 0.0143671 & 0.00651458 \\ 0 & 0 & 0.0143814 \end{pmatrix}$$

As previously we can estimate that the normal of these two planes have a component on the x axis and no component along the y axis. We have not found the last plane because only a few points are on this plane and we have to increase the number of iteration. If we set the number of iterations to 20000 iterations, we obtain as expected the three planes (see figure 24). For the last collineation, we obtain:

$$\mathbf{H_3} = \begin{pmatrix} 0.0209325 & -5.31735e-05 & -0.99922 \\ -2.68491e-05 & 0.0208353 & 0.0158142 \\ 0 & 0 & 0.0209152 \end{pmatrix}$$

Thus, we also obtained results which do not simply consists into a calibration grid ! However we must notice that this algorithm becomes a little heavy and asks a lot of calculus (something like 20000 iterations). In order to improve the speed of the algorithm we can use a parallel architecture. Such an architecture has been implemented, as desribed in appendix B, using the P.V.M. library.

In the case of real data, we could also use well known segmentation algorithms to initialize the sets of points used to estimate constraints, in order to improve efficiency. Moreover we could, using this new architecture, use concurrent algorithms or models to compute the structure of a scene. This will be the topic of a future study.

Figure 23: Estimation of planar structures - 1000 iterations

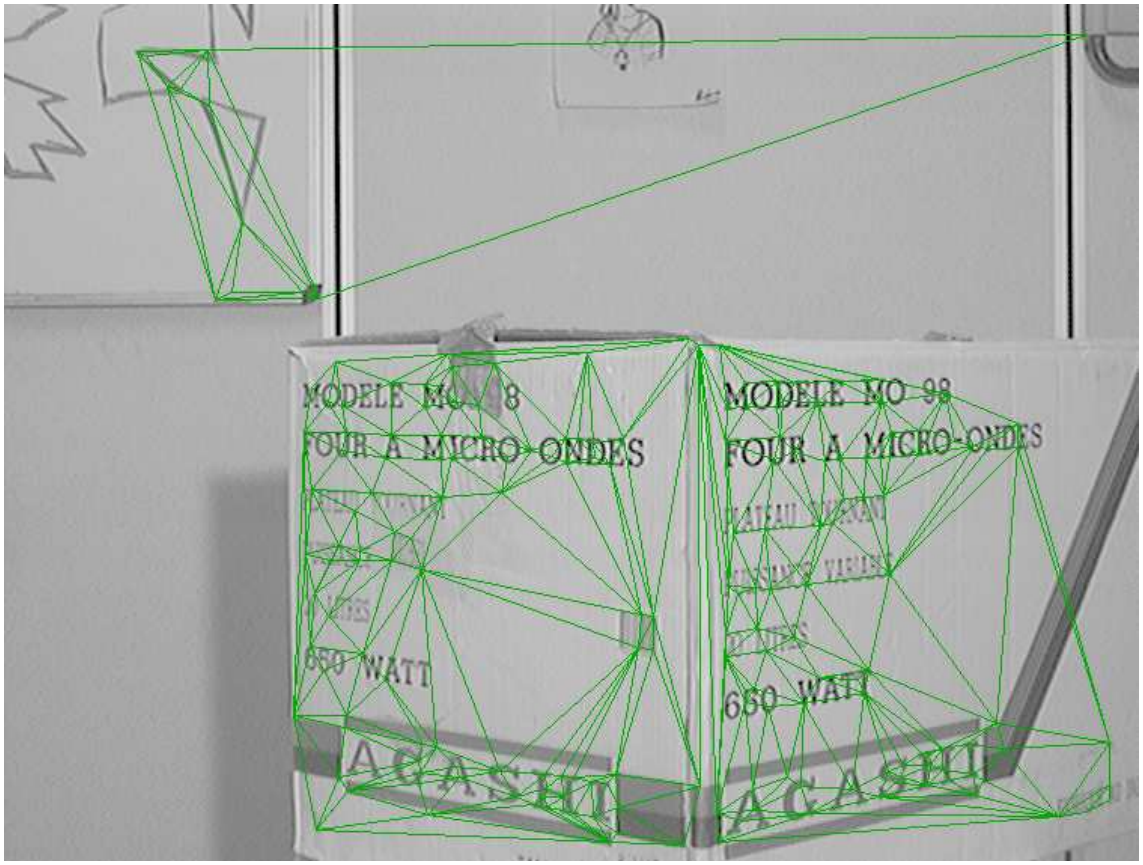

Figure 24: Estimation of planar structures - 20000 iterations

6 Conclusion

We have reviewed and completed the description of a general framework which allows not only to estimate a parameterization of the rigid displacement between two frames but also to determine several particular cases which occur in practice and have important advantages with respect to the scene structure analysis.

The statistical framework proposed to implement these equations is far from being new and is now of common use in the field of computer vision. However, the algorithm has been somehow modified to integrate two new aspects: (i) clustering data and (ii) testing different models to represent the data. Moreover, the specificity of the equations have been used to ease the implementation of least-square minimization.

The algorithm may be easily improved by using several different algorithms and different initializations types of the random subsets, in different parallel tasks. All the parallel structure have been yet implemented.

The implementation of this module itself is quite huge but still at the scale of a single software module which can be embedded in a high-level Image Understanding Environment.

References

[1] Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *Int'l J. Comp. Vision*, 7:333–356, 1988.

[2] R. Bajcsy. Active perception. *Proc IEEE 76*, 8:996–1005, 1988.

[3] D. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

[4] C. M. Brown. Gaze controls with interactions and delays. Technical report, OUEL: 1770/89, 1989.

[5] F. Chaumette. *La commande référencée vision*. PhD thesis, University of Rennes, Dept of Comp. Science, 1990. PhD thesis.

[6] F. Chaumette and S. Boukir. Structure from motion using an active vision paradigm. In *11th Int. Conf. on Pattern Recognition, The Hague, Netherlands*, 1991.

[7] B. Espiau. Effect of camera calibration errors on visual servoing in robotics. In *3rd International Symposium on Experimental Robotics, Kyoto,japan*, 1993.

[8] P. E. D. S. Facao, F. Romann, and T. Viéville. Couplage inertie vision pour un navigateur autonome. Technical Report Rapport de Recherche No 86J0326, Direction des Recherches Etudes et Techniques du Ministère de la Défense, 1992.

[9] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig ? In *2nd ECCV*, Genova, 1992.

[10] O. Faugeras. *Three-dimensional Computer Vision: a geometric viewpoint*. MIT Press, Boston, 1993.

[11] O. Faugeras, Q. T. Luong, and S. Maybank. Camera self-calibration : Theory and experiment. In *Second European Conference on Computer Vision*, Genoa, 1992.

[12] E. Francois and P. Bouthemy. Multiframe-based identification of mobile components of a scene with a moving camera. In *Conf. Computer Vision and Pattern Recognition, Hawai*, pages 166–172. IEEE Computer Society Press, Alamitos, Califiornia, 1991.

[13] B. Giai-Checa, P. Bouthemy, and T. Viéville. Detection of moving objects. Technical Report RR-1906, INRIA, Sophia, France, 1993.

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings Alvey Conference*, pages 189–192, 1988.

[15] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 761–764, 1992.

[16] J. Lavest, G. Rives, and M. Dhome. 3D reconstruction by zooming. In *Intelligent Autonomous System, Pittsburg*, 1993.

[17] Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. In *3rd E.C.C.V., Stockholm*, 1994.

[18] T. Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement.* PhD thesis, Université de Paris-Sud, Orsay, 1992.

[19] J. M. Martínez and L. Montano. A camera motion strategy to localize uncertain 3D lines. In *1993 IEEE International Conference on Systems, Man and Cybernetics*, pages 517–522, Le Touquet-France, October 1993.

[20] D. Murray, P. MacLauchlan, I. Reid, and P. Sharkey. Reactions to peripheral image motion using a head/eye platform. In *4th ICCV*, pages 403–411. IEEE Society, 1993.

[21] K. Pahlavan, J.-O. Ekhlund, and T. Uhlin. Integrating primary ocular processes. In *2nd ECCV*, pages 526–541. Springer Verlag, 1992.

[22] K. Pahlavan, T. Uhlin, and J.-O. Ekhlund. Dynamic fixation. In *4th ICCV*, pages 412–419. IEEE Society, 1993.

[23] N. Papanikolopoulos, B. Nelson, and P. Khosla. Full 3D tracking using the controlled active vision paradigm. In *The 7th IEEE Symposium on Intelligent Control, Glasgow, August*, 1992.

[24] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical recipes, the art of scientific computing.* Cambridge University Press, Cambridge, U.S.A., 1988.

[25] L. Robert and O. Faugeras. Relative 3d positionning and 3d convex hull computation from a weakly calibrated stereo pair. In H. Nagel, editor, *4th I.C.C.V., Berlin*. IEEE Computer Society Press, Los Alamitos, California, 1993.

[26] L. Robert and O. Faugeras. Relative 3-D positioning and 3-D convex hull computation from a weakly calibrated stereo pair. *Image and Vision Computing*, 13(3):189–197, 1995. also INRIA Technical Report 2349.

[27] J. K. S. Wu. A gradient-based method for general motion estimation and segmentation. *Journal of Visual Communication and Image Representation*, 4(1):25–38, 1993.

[28] L. Shapiro and M. Brady. Rejecting outliers and estimating errors in an orthogonal regression framework. Tech.Report OUEL 1974/93, Dept. Engineering Science, University of Oxford, Feb. 1993.

[29] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, 1994.

[30] S. Sull and N. Ahuja. Segmentation, matching and estimation of structure-and-motion of textured piecewise planar surfaces. In *IEEE Workshop on Visual Motion, Princeton, October*, pages 274–279, 1992.

[31] A. K. Tarabanis and P. A. an dR.Y. Tsai. A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1), 1995.

[32] W. Thompson and T. Pong. Detecting moving objects. *Int. Journal of Computer Vision*, 4:39–57, 1990.

[33] P. Torr and D. W. Murray. Stochastic motion clustering. In *The Third European Conference on Computer Vision*. Springer Verlag, Berlin, 1994.

[34] T. Viéville. Autocalibration of visual sensor parameters on a robotic head. *Image and Vision Computing*, 12, 1994.

[35] T. Viéville. *A few steps towards 3D Active Vision*. Springer Series in Information Sciences, 1996. To appear.

[36] T. Viéville, E. Clergue, R. Enciso, and H. Mathieu. Experimentating with 3D vision on a robotic head. *Robotics and Autonomous Systems*, 14(1), 1995.

[37] T. Viéville, P. Facao, and E. Clergue. Computation of ego-motion using the vertical cue. *Machine Vision and Applications*, 8, 1995.

[38] T. Viéville and O. Faugeras. The first order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 1995. To appear.

[39] T. Viéville and D. Lingrand. Using singular displacements for uncalibrated monocular visual systems. Technical Report RR-2678, INRIA, 1995.

[40] T. Viéville, Q. Luong, and O. Faugeras. Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision*, 17:1, 1996.

[41] T. Viéville, C. Zeller, and L. Robert. Using collineations to compute motion and structure in an uncalibrated image sequence. *International Journal of Computer Vision*, 18:2, 1996.

[42] R. Willson. *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, Department of Electical and Computer Engineering, Carnegie Mellon University, 1994.

[43] R. Willson and S. Shafer. What is the center of the image ? In *IEEE Proc CVPR'93, New-York, June*, pages 670–671, 1993.

[44] K. Wohn and A. Waxman. The analytic structure of image flows: deformation and segmentation. *Computer Vision Analysis and Image Processing*, 49:127–151, 1990.

[45] C. Zeller and O. Faugeras. Applications of non-metric vision to some visual guided tasks. In *Proceedings of the International Conference on Pattern Recognition*, pages 132–136, Jerusalem, Israel, Oct. 1994. Computer Society Press. A longer version in INRIA Tech Report RR2308.

[46] Z. Zhang, R. Deriche, Q.-T. Luong, and O. Faugeras. A robust approach to image matching: Recovery of the epipolar geometry. In *Proc. International Symposium of Young Investigators on Information\Computer\Control*, pages 7–28, Beijing, China, Feb. 1994.

A Estimation of the motion parameters.

A.1 Estimating the F-matrix

As discussed in [46] an efficient criterion is the average retinal Euclidean distances between each point $\mathbf{m}'$ and its epipolar line, defined by $\mathbf{F}\,\mathbf{m}$. The following symmetric constrained least-square criterion is minimized:

$$\begin{cases} \mathbf{F}_\bullet = argmin_{\mathbf{F}} \underbrace{\left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \underbrace{\left[d(\mathbf{m}', \mathbf{F}\mathbf{m})^2 + d(\mathbf{m}, \mathbf{F}^{\mathbf{T}}\mathbf{m}')^2\right]}_{f_{\mathbf{m}}(\mathbf{F})^2}\right] / \left[2 \sum_{\{\mathbf{m}\}} w_{\mathbf{m}}\right]}_{[\epsilon_{\mathbf{F}}(\mathbf{F})]^2} \\ ||\mathbf{F}_\bullet|| = 1, det(\mathbf{F}_\bullet) = 0 \end{cases} \quad (34)$$

where $w_{\mathbf{m}}$ is a weighted corresponding to the precision of the match, in fact the inverse of the variance of the precision of the match. The quantity $w_{\mathbf{m}}$ is given in $pixel^{-2}$, while $\epsilon_{\mathbf{F}}(\mathbf{F})$, the *average distance to the epipolar*, is in pixel.

The previous criterion can also be written as:

$$\mathbf{F}_\bullet = argmin_{\mathbf{F}} \underbrace{\sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \underbrace{\left[\lambda_{\mathbf{m}}(\mathbf{F})\ (\mathbf{g}_{\mathbf{m}}^T\ \mathbf{F})^2\right]}_{f_{\mathbf{m}}(\mathbf{F})^2} \Big/ \left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}}\right]}_{[\epsilon_{\mathbf{F}}(\mathbf{F})]^2} \tag{35}$$

with: $(\mathbf{g}_{\mathbf{m}}^T\ \mathbf{F}) = \mathbf{m}'^T \mathbf{F}\mathbf{m}$ and $\lambda_{\mathbf{m}}(\mathbf{F}) = \frac{1}{((\mathbf{F}\,\mathbf{m})^0)^2+((\mathbf{F}\,\mathbf{m})^1)^2} + \frac{1}{((\mathbf{F}^T\,\mathbf{m}')^0)^2+((\mathbf{F}^T\,\mathbf{m}')^1)^2}$.

Having this particular form we can easily derive an algorithmic schema, equivalent to usual technic of minimization of a non-linear least-square criterion [24], but much easier to implement. Let us consider we have an initial estimate $\bar{\mathbf{F}}$ of the parameter $\mathbf{F}$. At each step we minimize the following modified criterion:

$$\mathbf{F}_\bullet = argmin_{\mathbf{x}} \underbrace{\sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \left[\lambda_{\mathbf{m}}(\bar{\mathbf{F}})\ (\mathbf{g}_{\mathbf{m}}^T\ \mathbf{F})^2\right] \Big/ \left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}}\right]}_{[\epsilon_{\mathbf{F}}(\mathbf{F})]^2} \tag{36}$$

which is now a quadratic criterion which solution is given by the following normal equations:

$$\underbrace{\sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \lambda_{\mathbf{m}}(\bar{\mathbf{F}})\ \mathbf{g}_{\mathbf{m}}\ \mathbf{g}_{\mathbf{m}}^T}_{\mathbf{M}(\bar{\mathbf{F}})}\ \mathbf{F} = 0 \tag{37}$$

for the homogeneous quantity $\mathbf{F}$.

We fix the highest component of $\bar{\mathbf{F}}$ to 1 in $\mathbf{F}$ and solve the linear system with respect to the others variables.

Furthermore given a matrix $\mathbf{F}$ it is always trivial to compute the "closest matrix" $\mathbf{F}$ with $||\mathbf{F}|| = 1$[8] and the "closest matrix" $\mathbf{F}$ with $det(\mathbf{F}) = 0$[9].

[8] Just take $\bar{\mathbf{F}} = \frac{\mathbf{F}}{||\mathbf{F}||}$.

[9] Let us write :

$$\mathbf{F} = \left[\underbrace{(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)}_{\mathbf{U}}\right] \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \left[\underbrace{(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)}_{\mathbf{V}}\right]^T \tag{38}$$

with $\mathbf{U}\,\mathbf{U}^T = \mathbf{U}^T\,\mathbf{U} = \mathbf{I}$, $\mathbf{V}\,\mathbf{V}^T = \mathbf{V}^T\,\mathbf{V} = \mathbf{I}$ and $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. This decomposition, called *singular value decomposition*, is unique and can be efficiently estimated numerically [24]. The reader can easily check that the matrix $\bar{\mathbf{F}} = \mathbf{U} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{V}^T$ verifies $det(\bar{\mathbf{F}}) = 0$ and minimizes $||\mathbf{F} - \bar{\mathbf{F}}||$ considering the norm $||\mathbf{F}|| = sup_{||\mathbf{u}||=1}||\mathbf{F}\,\mathbf{u}||$. Moreover $\mathbf{u}_3$ is the vector generating the kernel of $\bar{\mathbf{F}}^T$, i.e. $\mathbf{u}_3^T\,\bar{\mathbf{F}} = 0$ and $\mathbf{v}_3$ is the vector generating the kernel of $\bar{\mathbf{F}}$, i.e. $\bar{\mathbf{F}}\,\mathbf{v}_3 = 0$.

Therefore this criterion can be recursively minimized by iterating the following process:
(1) Compute $\mathbf{F}$ using equation (37).
(2) Reproject $\mathbf{F}$ in order to verify the required constraints.

If we do not have an initial estimate $\bar{\mathbf{F}}$ of the parameter, we can take $\lambda_{\mathbf{m}}(\bar{\mathbf{F}}) = 1$.

Here, we need at least 7 non-coplanar points, since the F-matrix is defined by 7 parameters.

A.2 Using a robust estimation method.

In fact, when considering matches, we may find both bad locations and false matches, or matches which do not correspond to the quantity we want to estimate, because they belongs to moving objects. In order to eliminate these outliers, and following previous authors in this field [28, 46] we use a variant of the least-median square algorithm, described now.

A Monte Carlo type technique is used to draw random subsamples of different point correspondences and for each subsample we determine the F-matrix .from equation (36). Then we compute the residual error $f_{\mathbf{m}}(\mathbf{F}_\bullet)^2$ for all matches, and detect the matches which residual error is lower than a given threshold ϵ_* in pixel (typically 1 or 2 pixels), i.e. compute $w_{\mathbf{m}}$ as:

$$w_{\mathbf{m}} = \begin{cases} 1 & if\, f_{\mathbf{m}}(\mathbf{F}_\bullet) \leq \epsilon_* \\ 0 & otherwise \end{cases} \tag{39}$$

Let $N_{\mathbf{F}_\bullet}$ be the number of points with $w_{\mathbf{m}} = 1$, i.e. which estimation is compatible with the estimation of $\mathbf{F}_\bullet$.

We repeat this mechanism and finally select as "best" estimate the one for which a maximal number of points is compatible with the given estimate. This method can be summarized as follows :

$$\mathbf{F}_\star = argmax_{\mathbf{x}} N_{\mathbf{F}_\bullet} \tag{40}$$

since the algorithm attempt to maximize the number of elements for each class. Moreover the following heuristics are used:

- If two set of points have the same elements, we merge them and retain the parameter for which the average residual error is minimal to represent the new class.
- If two set of points have similar parameters we can also merge them.

It turns out that this method is very robust to false matches as well as outliers due to bad locations. The obtained estimate is refined, at last, solving a least-squares problem, as discussed previously. This method, is of exponential

complexity[10]. Such exponential complexity is always present when using robust method eliminating outliers, but can be reduced by several heuristics, as proposed here.

The previous algorithm can also be used to detect *several simultaneous motions*, since after the detection of the "main" displacement, the same algorithm can be applied again on the matches not already selected and which are expected to be detected during this second steps among others outliers. Such a variant of this mechanism has been already experimented to distinguish objects in motion, using a particular form of the F-matrix as reported in [33] and discussed in the next section. Moreover, using this kind of statistical tests, several heuristics have been developed to select a set of samples which likely corresponds to objects with coherent retinal motion [13].

As a conclusion, following this general method, we segment moving objects by simply considering all data points which define a rigid displacement.

A.3 Estimation of planar structures.

Following the same method as for the F-matrix an efficient criterion is to minimize the residual disparity again, as in [41] and obtain $\mathbf{H}$ through:

$$\left\{ \begin{array}{l} \mathbf{H}_\bullet = argmin_{\mathbf{H}} \underbrace{\left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \underbrace{||\mathbf{m}' - \frac{\mathbf{H}\,\mathbf{m}}{((\mathbf{h}^2)^T\,\mathbf{m})}||^2}_{f_{\mathbf{m}}(\mathbf{H})^2} \right] / \left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}} \right]}_{[\epsilon_{\mathbf{H}}(\mathbf{H})]^2} \\ H_\bullet^{22} = 1 \end{array} \right. \tag{41}$$

[10] We can easily analyze the complexity of the algorithm. Let us consider that the F-matrix is a vectorial parameter $\mathbf{x}$.

Let us assume that we do have Q set of points corresponding to parameters $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_Q\}$. These parameters correspond to Q classes of data points with $\{N_1, N_2, ...N_Q\}$ elements respectively, which has been mixed together to form a set of N data points. When we select one point, there is a probability $p_i = \frac{N_i}{N}$ that it belongs to a class of index i, when we select $dim(\mathbf{x})$ points, there is thus a probability $P_i = \frac{N_i}{N} \frac{N_i-1}{N-1} ... \frac{N_i-dim(\mathbf{x})}{N-dim(\mathbf{x})}$ that they all belong to the same class of index i, so that when we select $dim(\mathbf{x})$ points, the probability to fail into one among the Q classes, i.e. to have a coherent selection is $P = \sum_{i=1}^{Q} P_i$. Finally the probability to obtain a successful selection of samples, i.e. to have a coherent selection for the Q classes, after $\sum_{i=0}^{Q} N_i$ trials is :

$$P = \prod_{f=0}^{Q-1} \sum_{q=1}^{Q-f} \prod_{k=0}^{dim(\mathbf{x})-1} \left[\frac{N_i - k}{[\sum_{j=1}^{Q-f} N_j] - k} \right] \simeq \frac{1}{Q^{dim(\mathbf{x})\,Q}}$$

where we write $\mathbf{H} = (\mathbf{h}^0, \mathbf{h}^1, \mathbf{h}^2)$ in order to use this compact notation[11].

For equation (41) we can write: $(\mathbf{g}_\mathbf{m}^T\ \mathbf{H}) = ((\mathbf{h}^2)^T\ \mathbf{m})\ \mathbf{m}' - \mathbf{H}\ \mathbf{m}$ and $\lambda_\mathbf{m}(\mathbf{H}) = \frac{1}{(\mathbf{h}^2)^T\ \mathbf{m}}$, and use the same mechanism of estimation.

We need at least 4 non-collinear points, since a H-matrix is a 3x3 matrix up to a scale factor thus defined by 8 parameters, while each match provides 2 equations. However if we introduce the constraint that the collineation is compatible with a rigid displacement, only at least 3 points are needed as discussed previously.

The error $\epsilon_\mathbf{H}(\mathbf{H})$, given in pixel, will be called *residual disparity after motion reduction* in the sequel.

Moreover we can use such an estimation to estimate a collineation in relation with a fundamental matrix. In this case, we have just to estimate the vector $\mathbf{h}$ of equation (12). Following [41], for our three points we can write:

$$\mathbf{h}^T \mathbf{m_i} = \pi_{\mathbf{m_i}}$$

and compute $\mathbf{h}$ as:

$$\mathbf{h} = \left(\mathbf{m_1}^T \mathbf{m_2}^T \mathbf{m_3}^T\right)^{-1} (\pi_{\mathbf{m_1}} \pi_{\mathbf{m_2}} \pi_{\mathbf{m_3}})$$

Moreover we can also compute $\mathbf{h}$ from $\mathbf{H}$ and $\mathbf{F}$ using equation (12) in order to use criterion (41) to estimate a collineation compatible with a fundamental matrix.

We thus can compute collineations from correspondences but also from a set of correspondences and a fundamental matrix.

A.4 Grouping collinear points.

Given a set of points, if they are collinear, they always constitute a singular configuration for the previous estimations. We therefore, will apply the same method again to group collinear structures.

This is done using the following quadratic criterion:

$$\begin{cases} \mathbf{l}_\bullet = argmin_\mathbf{l} \underbrace{\left[\sum_{\{\mathbf{m}\}} w_\mathbf{m}\ \underbrace{(\mathbf{l}^T\ \mathbf{m})}_{f_\mathbf{m}(\mathbf{l})^2}\right] / \left[\sum_{\{\mathbf{m}\}} w_\mathbf{m}\right]}_{[\epsilon_\mathbf{l}(\mathbf{l})]^2} \\ (l^0_\bullet)^2 + (l^1_\bullet)^2 = 1 \end{cases} \tag{42}$$

[11] The relation $\mathbf{m}' = \frac{\mathbf{H}\ \mathbf{m}}{((\mathbf{h}^2)^T\ \mathbf{m})}$ is a vectorial form for :

$$\begin{cases} u' &= \frac{H^{00}\ u + H^{01}\ v + H^{02}}{H^{20}\ u + H^{21}\ v + H^{22}} \\ v' &= \frac{H^{10}\ u + H^{11}\ v + H^{12}}{H^{20}\ u + H^{21}\ v + H^{22}} \\ 1 &= 1 \end{cases}$$

and the same mechanism as described before allows to compute the $\mathbf{l}$, in one step since we have linear normal equations, i.e. $\lambda_{\mathbf{m}}(\mathbf{l}) = 1$ and $g_{\mathbf{m}}(\mathbf{l}) = f_{\mathbf{m}}(\mathbf{l})$.

As well known, the vector $\mathbf{l} = (l^0, l^1, l^2)$ represents the line, and has been normalized in order $|f_{\mathbf{m}}(\mathbf{l})|$ to corresponds to the retinal distance from the point $\mathbf{m}$ to the line $\mathbf{l}$.

Again the proposed formalism allows to detect collinear structures at the same time their parameters are estimated.

As soon as we have decided that a set of points is collinear, we can choose two points in the set, say the two extremities of the line-segment which contains all the points (the convex hull of the points in fact) and carry on all subsequent computations using these two points only while other points only provide a redundant information.

Another important property is that *their order on the line is preserved through the projection*, as soon as this line is a visible line [25].

Moreover, the chosen points will be reprojected on the line, in order to increase the precision of the estimation of their location.

Note that we compute bidimentional lines and not lines in space: A projection preserves the alignement of points but also create erroneous collinear points. Following [41], we can use the following algorithm to determine true collinear points of the space:

- Select three points collinear in the first image, m_1, m_2 and m_3.
- If (m_1, m_2) corresponds to an epipolar line, the points $\mathbf{M_1}$, $\mathbf{M_2}$ and $\mathbf{M_3}$ are collinear if and only if $\{e, m_1, m_2, m_3\} = \{e', m'_1, m'_2, m'_3\}$.
- If (m_1, m_2) is not an epipolar line, the points are collinear if and only if m'_1, m'_2 and m'_3 are collinear.

This allows to estimate collinear points in the plane which are collinear in space.

A.5 Depth and calibration fusion along an image sequence.

Let us now consider we have more than two frames, i.e. an image sequence. Using equation (6) in the next frame, we obtain:

$$\frac{||\mathbf{s}'||}{Z'} + (\mathbf{r}'^T \mathbf{m}') = \pi'_{\mathbf{m}'} \tag{43}$$

and combining equations (6), (8) and (43) we obtain another estimate of the depth:

$$\nu_{\mathbf{m}} \frac{||\mathbf{s}||}{Z'} + (\mathbf{r}^T \mathbf{m}) = \pi_{\mathbf{m}} \tag{44}$$

and a relation between $\mathbf{r}$, $\frac{||\mathbf{s}||}{||\mathbf{s}'||}$ and $\mathbf{r}'$:

$$\epsilon_{\mathbf{m},\mathbf{r}'}(\mathbf{r}, \frac{||\mathbf{s}||}{||\mathbf{s}'||}) = \pi_{\mathbf{m}} - \mathbf{m}^T\ \mathbf{r} + \nu_{\mathbf{m}}\, \mathbf{m}'^T \left[\frac{||\mathbf{s}||}{||\mathbf{s}'||}\mathbf{r}'\right] - \pi'_{\mathbf{m}'}\, \nu_{\mathbf{m}}\, \frac{||\mathbf{s}||}{||\mathbf{s}'||} = 0 \qquad (45)$$

where $\epsilon_{\mathbf{m},\mathbf{r}'}(\mathbf{r}, \frac{||\mathbf{s}||}{||\mathbf{s}'||})$ is given in pixel.

Using these equations:

1. We can estimate a value of $\mathbf{r}$ and $\frac{||\mathbf{s}||}{||\mathbf{s}'||}$, knowing $\mathbf{r}'$, using equation (45) for at least four non-collinear points, with general values of depth. This can be obtained minimizing the following quadratic criterion:

$$(\mathbf{r}_\bullet, \frac{||\mathbf{s}||}{||\mathbf{s}'||}_\bullet) = argmin_{(\mathbf{r}, \frac{||\mathbf{s}||}{||\mathbf{s}'||})} \underbrace{\left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}}\ \underbrace{\epsilon_{\mathbf{m},\mathbf{r}'}(\mathbf{r}, \frac{||\mathbf{s}||}{||\mathbf{s}'||})^2}_{f_{\mathbf{m}}(\mathbf{r}, \frac{||\mathbf{s}||}{||\mathbf{s}'||})^2}\right] / \left[\sum_{\{\mathbf{m}\}} w_{\mathbf{m}}\right]}_{\left[\epsilon_{\mathbf{r}}(\mathbf{r}, \frac{||\mathbf{s}||}{||\mathbf{s}'||})\right]^2} \qquad (46)$$

   $w_{\mathbf{m}}$ being defined as in equation (34). The error $\epsilon_{\mathbf{r}}(\mathbf{r}, \frac{||\mathbf{s}||}{||\mathbf{s}'||})$, given in pixel, will be called *residual fusion disparity* in the sequel.

   The same mechanism as described before allows to compute the estimate, in one step since we have linear normal equations.

   This equation will also be used for auto-calibration as discussed in the next section.

2. We can estimate the depth by fusing the value predicted by equation (43) and (44), i.e. from information in the present frame and the previous frame.

In the case where we take $\mathbf{r}' = 0$ this allows to estimate $\pi'_{\mathbf{m}'}$ from two values.

B Using P.V.M. for a parallel approach

In order to build an efficient algorithm, we can use the software standard P.V.M. It consists in a library which enables to use heterogeneous computers as a parallel machine with all the advantages of parallelism and the advantages of a high level programming library. All the messages and data exchanges (over a network which can contain several differents architectures) becomes transparent to the programmer which has just to design the software architecture of his application and to manage message exchanges.

For our application, we use P.V.M on a network of Sun workstations running under different operating systems. We can easily separate our algorithm in two part (figure 25):

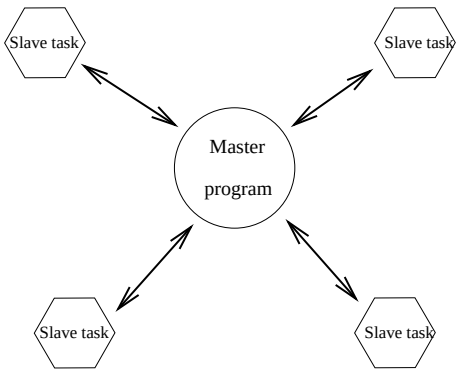

Figure 25: Parallel architecture chosen

- A main program, "the master", only select random sets of points and manage the distribution of constraints with respect to the hierarchy previously described in 4.2. Moreover this part of the program has to manage all the data and the scene dynamic representation.
- Several tasks, "the slaves", running on different processors perform computation for the previous program. Such a task only take data, estimate a constraint using these data and return the result to the master. Moreover we can note that the constraint to be estimated is chosen by the master program.

It's clear that this architecture is efficient if:

- an estimation is a task leading into a lot of computation,
- messages are as short as possible,
- the master is optimized in order to be able to send work to tasks as soon as they have finished their previous computations. The master must construct an efficient representation of the scene to be able to store and retrieve data quickly. Moreover this task has to respect the hierarchy and the stratified definition of constraints. We notice here an important fact: the representation of the scene may change between the moment where the master send a request to a slave and the moment where this slave answer. For instance, two slaves may find the same constraints, complementary informations, etc ...

We have chosen the following simple but surprisingly perhaps efficient heuristic to distribute the constraints to the slave tasks. For each constraint selected by the master:

- First we send an estimation request to each task,

- Then as we receive a result, we send another request to the task. We notice here that a result may be a positive result (results of the estimation plus a list of points, plus ...) but also a negative result (there is no results to this request). On both cases, the slave return a message in order to get another request from the master.

- When the master chooses to stop the estimation of this constraint, we wait for all the constraint to send their results.

This heuristic allows the master to balance the distribution of the work between the different tasks. If a processor is more efficient than the others, it will receive more requests.

C Generating matches along a image sequence

As many algorithms of this field of computer vision, the algorithm considers point correspondences as input. Available image matching programs, were not well adapted to our problem where we consider correspondences between the extreme views of an image sequence as described in 4.1.4 We thus developped a new approach for this problem of computing correspondences:

- We detect corner points in the first image.

- We track the correspondences between the images sequence, choosing the best method for our case.

- We return the correspondences detected in the first image and still present in the last image of the sequence.

We use a correlation algorithm to track the correspondences. In order to minimize disparity beetween two frames and thus facilitate the work of the correlation, we use a stabilization routine which predicts the position of a correspondence. Then we run the correlation routine on a stabilized pair of images, with a smaller search area. This stabilization matrix is computed to reduce the disparity between the two images by considering histograms of edges coordinates. The correlation routine and the stabilization routine are both extracted from the acv library[12].

The parameters are set in order to produce a lot of matches and to eliminate false matches during the sequence.

The results obtained using the image-matching program are presented in figure 26

[12] Acv is a library developped in the Robovis project and is designed for the implementation of adaptive mechanisms of reactive vision.

Figure 26: Using the image matching program: Correspondences between two images

If we use our method between only two images, we verify easily that the correspondences are many but a lot a false correspondences are estimated: see figure 27

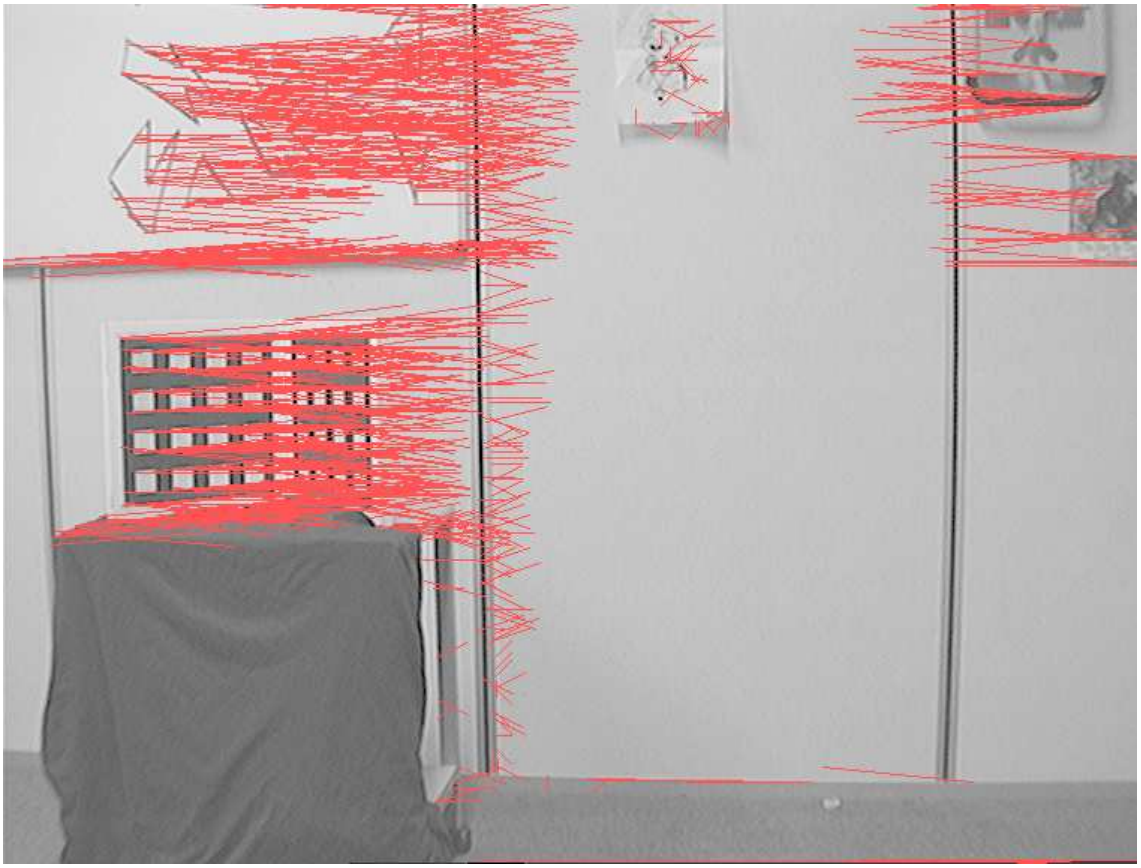

Figure 27: Correlation between two images

In order to analyze easily the results, we represent the matches using a scale factor of 10. Now if we track correspondences in a sequence of 10 images and represent the correspondences obtained for the first and the last views, we obtain the results presented in figure 28

As expected, we obtain many correspondences, whereas most of the false correspondences are eliminated. This results seems to be better, comparing to actual RobotVis softwares.

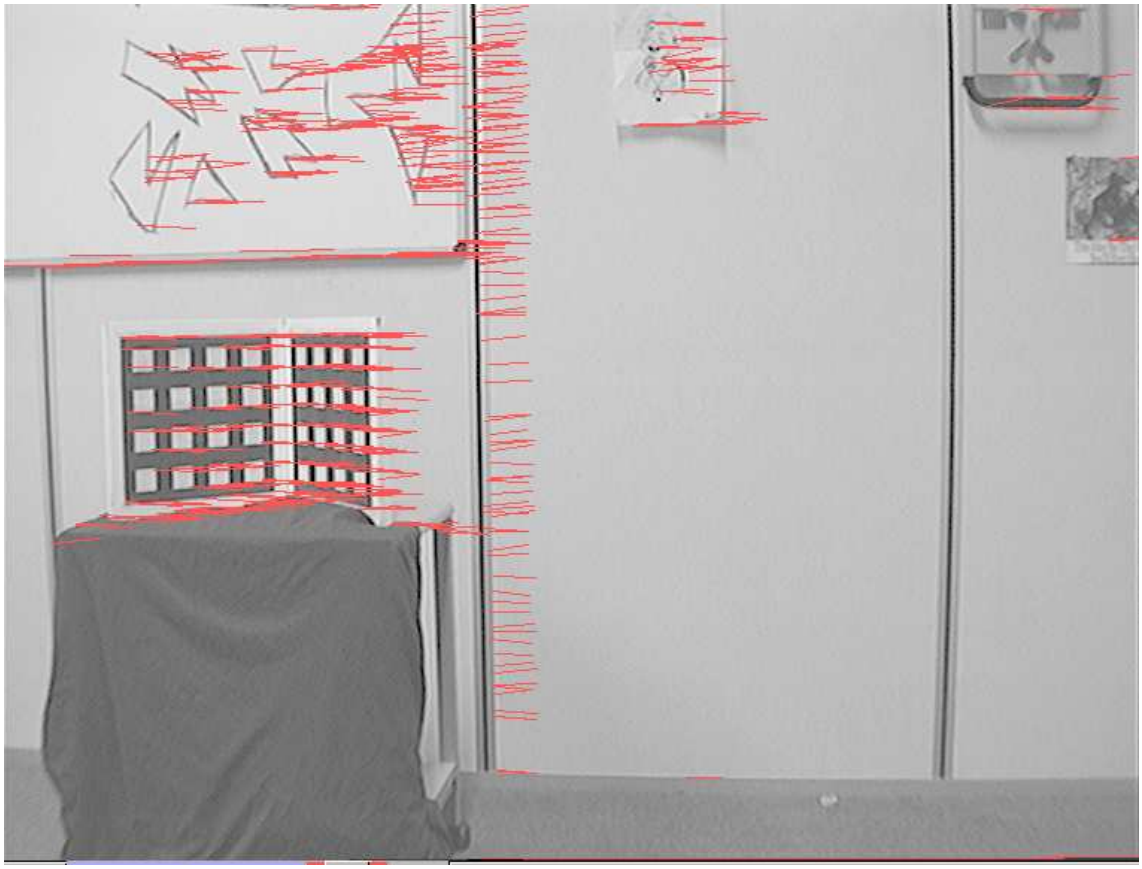

Figure 28: Tracking correspondences along an image sequence

D Why we've choosen to work in the uncalibrated case

Let us work on a very fundamental and well known problem : calibration. We are looking at a calibration grid from different places and we want to compute the extrinsic parameters at each time. We use the same robotic system that we used for our experimental results and perform only a pure translation on the x axis (auto-focus and zoom are fixed). The position of the optic center is taken from the position of the calibration grid as reference. We obtain the following results for a "high amplitude" sequence:

image number	T_x (mm)	Position of the optic center
1	0	$C_0 = \begin{pmatrix} -959.116 & -77.5274 & -1738.3 \end{pmatrix}^T$
2	120	$C_0 = \begin{pmatrix} -1052.2 & -77.7839 & -1669.19 \end{pmatrix}^T$
3	220	$C_0 = \begin{pmatrix} -1092.08 & -75.5906 & -1559.7 \end{pmatrix}^T$
4	320	$C_0 = \begin{pmatrix} -1140.84 & -71.1244 & -1464.87 \end{pmatrix}^T$
5	420	$C_0 = \begin{pmatrix} -1204.93 & -71.4061 & -1397.43 \end{pmatrix}^T$
6	620	$C_0 = \begin{pmatrix} -1348.92 & -82.0162 & -1387.56 \end{pmatrix}^T$

We then can trace the evolution of optic center position in function of the translation performed in figure 29.

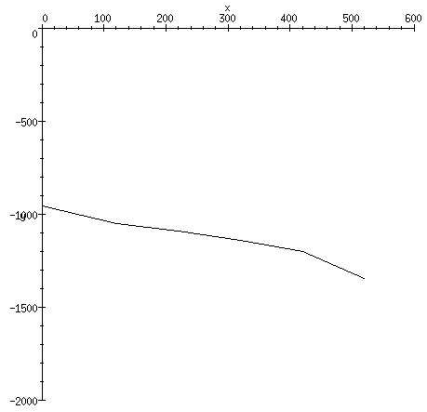

Figure 29: Estimation of the optic center position

We notice that the optic center position decreases when the camera comes close to the grid as expected. Moreover we can observe that this evolution corresponds to an important translation

But what happens if we try to estimate small variation of these parameters ? In fact we want to analyze the variation of the optic center when we are zooming like in [42], where these variations have been studied for a few cameras.

Thanks to Willson [42], we report here with his permission, his results concerning the position of the optic center during the zoom displacement: see figure 30 and figure 31.

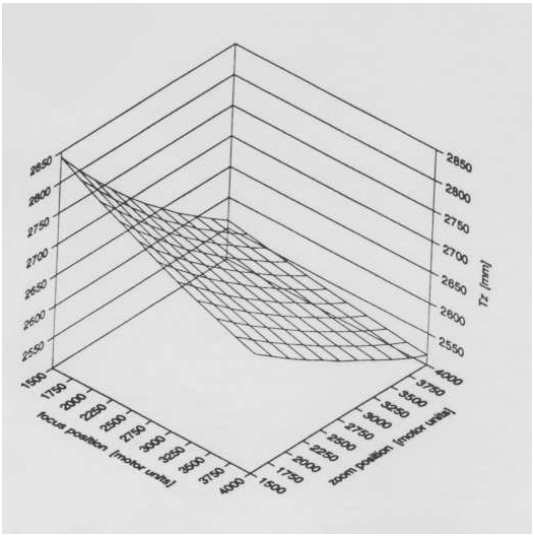

Figure 30: Estimation of T_z versus focus and zoom, from [42] with permission

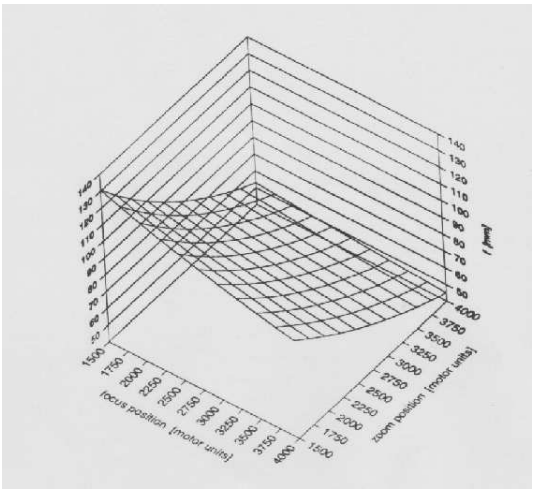

Figure 31: Estimation of f versus focus and zoom, from [42] with permission

Then we perform a smaller displacement:

image number	T_x (mm)	Position of the optic center
1	570	$C_0 = \begin{pmatrix} -1298.54 & -72.45 & -1278.91 \end{pmatrix}^T$
2	560	$C_0 = \begin{pmatrix} -1254.98 & -68.7061 & -1257.4 \end{pmatrix}^T$
3	550	$C_0 = \begin{pmatrix} -1258.26 & -68.0425 & -1266.69 \end{pmatrix}^T$
4	540	$C_0 = \begin{pmatrix} -1248.59 & -68.3305 & -1273.87 \end{pmatrix}^T$
5	530	$C_0 = \begin{pmatrix} -1263.27 & -71.088 & -1301.99 \end{pmatrix}^T$
6	520	$C_0 = \begin{pmatrix} -1253.44 & -72.7102 & -1326.23 \end{pmatrix}^T$

We can also trace the optic center evolution and we observe in figure 32 that we can't deduct anything from these results : the calibration lack of precision (10 % according to [36]) is greater than the displacement that we want to measure !

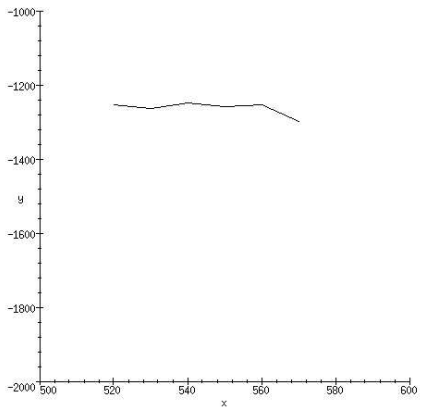

Figure 32: Estimation of the optic center position

On intrinsic parameters, other problems occur. For instance, we can record variations of 100 pixel for the principal point (on a 768×576 image) during a zoom (see 33).

As a conclusion, we have to find a new way to analyze small displacements. Moreover we must not not consider that the system is calibrated and that its intrinsic parameters are constant.

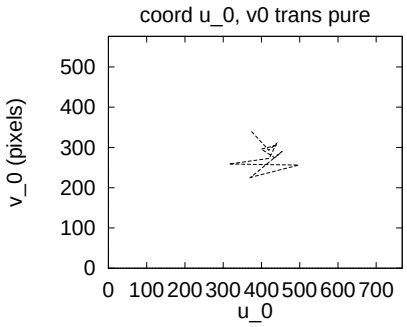

Figure 33: Estimation of the principal point during a zoom

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399